AD-A064 289 SCIENCE APPLICATIONS INC MCLEAN VA F/G 1/3
PREDESIGN OF THE SECOND GENERATION COMPREHENSIVE HELICOPTER ANA--ETC(U)
DEC 78 T HAMRICK, D COPELAND, F TARZANIN DAAJ02-77-C-0059
UNCLASSIFIED USARTL-TR-78-42 NL

1 OF 2
AD
A064 289

USARTL-TR-78-42

LEVEL (12)

PREDESIGN OF THE SECOND GENERATION COMPREHENSIVE HELICOPTER ANALYSIS SYSTEM

T. Hamrick, D. Copland, F. Tarzanin,
J. Staley, L. Hunt, and G. Burns

Science Applications, Inc.
1651 Old Meadow Road
McLean, Virginia 22101

December 1978

Final Report for Period September 1977 - May 1978

Approved for public release;
distribution unlimited.

Prepared for

APPLIED TECHNOLOGY LABORATORY
U. S. ARMY RESEARCH AND TECHNOLOGY LABORATORIES (AVRADCOM)
Fort Eustis, Va. 23604

APPLIED TECHNOLOGY LABORATORY POSITION STATEMENT

This report summarizes the results of the predesign phase of the Second Generation Comprehensive Helicopter Analysis System. The predesign phase was conducted to provide: improvements to the Government-written functional specification, conceptual system design, definition of necessary computer program configuration items, development specifications, and a baseline development plan. This phase will be subsequently followed by the development, validation, maintenance, and user application phases. The report is considered to be technically sound.

Technical program direction was provided by Messrs. P. H. Mirick and D. J. Merkley, Contracting Officer's Representatives (Technical), of the Applied Technology Laboratory; Mr. H. I. MacDonald, Team Leader; and Messrs. E. E. Austin, A. E. Ragosta, and W. D. Vann of the project team.

Unclassified
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: USARTL-TR-78-42	2. GOVT ACCESSION NO.
3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (and Subtitle): PREDESIGN OF THE SECOND GENERATION COMPREHENSIVE HELICOPTER ANALYSIS SYSTEM	5. TYPE OF REPORT & PERIOD COVERED: Final Report, Sep 1977 - May 1978
	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s): T. Hamrick, D. Copeland, F. Tarzanin, J. Staley, L. Hunt, and G. Burns	8. CONTRACT OR GRANT NUMBER(s): DAAJ02-77-C-0059
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Science Applications, Inc., 1651 Old Meadow Rd., McLean, VA 22101	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: 63211A 1L263211D157 18003 EK
11. CONTROLLING OFFICE NAME AND ADDRESS: Applied Technology Laboratory, U.S. Army Research and Technology Laboratories (AVRADCOM), Fort Eustis, VA 23604	12. REPORT DATE: December 1978
	13. NUMBER OF PAGES: 147
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)	15. SECURITY CLASS. (of this report): Unclassified
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Acoustics, Aerodynamic Loading, Aerodynamics, Aeroelasticity, Computer Programs, Control, Digital Simulation, Dynamics, Flight Simulation, Helicopter Instability, Loads (Forces), Mathematical Models, Noise, Performance, Rotary Wing Aircraft, Stability

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

A predesign study was conducted for a proposed Second Generation Comprehensive Helicopter Analysis System (CHAS). A draft Type A system specification was reviewed to determine its validity for a system to analyze helicopter performance, stability and control, loads, aeroelastic stability, and acoustics using consistent technology with a capability for several levels of complexity.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

Unclassified
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

79 02 02 073

408 404

A preliminary design for the system was developed to use for evaluation of the feasibility of CHAS. The resulting System consisted of an Executive and a set of Technology Modules to represent the technical aspects of the helicopter analysis problems. The preliminary System indicated that the draft Type A system specification requirements could generally be met by a plan to develop the System over approximately a 4-year period with a total professional development effort of approximately 80 man-years. The unique feature of the proposed System, which would permit meeting the comprehensive requirements of the system specification, is the concept of Model Builder/Model User developed under this study. This concept permits the engineer to "build" computer programs from Technology Modules developed for the System. The resulting programs (Specific Simulation Models) would contain only the technology needed by the engineer to solve his particular problem and would not require his program to carry the burden of all the general capability required by CHAS. The Science Applications/Boeing Vertol team conducting the study developed a software executive concept to the level of detail that the feasibility of the Model Builder/Model User concept was well established; technical requirements of the Technology Modules were established; and the feasibility of the Model Builder/Model User concept for using these modules to meet the requirements of the Type A system specification was tested and demonstrated by the Science Applications, Inc./Boeing Vertol predesign team.

PREFACE

This report summarizes the results of a study conducted for Applied Technology Laboratory (ATL), U. S. Army Research and Technology Laboratories (AVRADCOM), under Contract DAAJ02-77-C-0059. Technical program direction was provided by Messrs. Paul Mirick and Donald Merkley, Contracting Officer's Representatives (Technical) of ATL, Mr. H. I. MacDonald, Team Leader, and Messrs. E. E. Austin, A. E. Ragosta, and W. D. Vann of the project team.

The study reviewed a draft Type A system specification for the Second Generation Comprehensive Helicopter Analysis System (CHAS). CHAS is to have the capability to analyze helicopter performance, stability and control, loads, aeroelastic stability, and acoustics using analytical models with consistent technology having a capability for several levels of complexity. Important requirements for development of CHAS are accuracy, minimum operating cost, and user community acceptance. A preliminary design of a System was developed to use for evaluation of the feasibility of CHAS as defined by the requirements in the draft Type A system specification. The resulting System consisted of an Executive and a set of Technology Modules to represent the technical aspects of the helicopter analysis problems. The preliminary System indicated that the draft Type A system specification requirement could generally be met by a plan to develop the System over approximately a 4-year period with a total professional development effort of approximately 80 man-years. The unique feature of the proposed System that would permit meeting the system specification requirements is the concept of Model Builder/Model User, which permits the engineer to "build" computer programs from Technology Modules developed for the System. The resulting computer programs (Specific Simulation Models) would contain only the technology needed by the engineer to solve his problem and would not require his computer program to carry the burden of all the general capability required by CHAS.

The study was conducted as a joint effort between Science Applications, Inc. (SAI) of McLean, Virginia, and Boeing Vertol Co. (BV) of Philadelphia, Pennsylvania. SAI was the prime contractor and BV was a subcontractor. Participants for SAI were Dale Copeland, Director of Software Technology, Washington Operations, Thomas Hamrick, SAI Principal Investigator, Lee Hunt, Senior Software Analyst, and Gerald Burns, Software Analyst. BV participants were Frank Tarzanin, BV Project Manager, and James Staley, BV Project Engineer.

The SAI/BV team developed a software executive concept to the level of detail that the feasibility of the Model Builder/Model User concept was well established and the feasibility of the Model Builder/Model User concept for using these modules to meet the requirements of the Type A system specification was tested and demonstrated by the SAI/BV team.

TABLE OF CONTENTS

TABLE OF CONTENTS (continued)

TABLE OF CONTENTS (continued)

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS (continued)

LIST OF TABLES

EXECUTIVE SUMMARY

A predesign effort was conducted for development of the Second Generation Comprehensive Helicopter Analysis System (CHAS). This effort was part of a program plan for the development and application of CHAS.

A. WHAT IS THE SYSTEM?

The Science Applications, Inc./Boeing Vertol Company (SAI/BV) Second Generation Comprehensive Helicopter Analysis System is a system for creating simulation models for specific helicopter components, or for a complete helicopter system or subsystem. CHAS includes representation of the supporting functions required to allow the engineer to use the produced modules in a meaningful, cost beneficial manner.

The System creates these helicopter simulation modules at several levels of complexity which:

(1) Organize the equations of motion and data for the components of the simulated helicopter in a manner suited to determining a particular helicopter technical characteristic such as helicopter performance, and

(2) Define the procedures and methods needed for solutions of these equations which specify flight conditions.

The System is composed of:

(1) Technology Modules (TM):

- A TM is one or more interrelated Software Modules (SM) (sections of computer code) performing a singular processing function such as airloads, downwash, or acoustics. A Specific Technology Module (STM) is created by selecting one or more of the SMs in a TM.

(2) An Executive comprised of two parts:

- A model builder executive, which contains all functions required to produce a simulation model.

- A data handling executive which contains all functions required to validate input data and process output data.

(3) Specific Simulation Models (SSMs):

- An SSM is a simulation model produced by the Executive for a specific engineering study. There will be an SSM corresponding to each Particular Functional Capability (PFC) to be specified in the Type A system specification by the Applied Technology Laboratory (ATL).

The System has the capability of allowing the user to specify the desired coupling required for an SSM by defining a scenario of flow among the Technology Modules. The Scenario is defined in the Comprehensive Helicopter Analysis Research Language for Engineer Studies (CHARLES) developed by SAI/BV and includes the specifications of interruptions in the processing flow to examine intermediate results or to input new data parameters.

In summary, the SAI/BV CHAS is a system through which the user builds STMs from TMs and couples them by defining Scenarios of conditional flow. The System then produces an SSM which may be executed by an engineer studying a helicopter analysis program. Finally, the System produces the model output to present the results in a form most beneficial to the user.

B. HOW THE SYSTEM WORKS

The System consists of three major functional phases: model building, model execution, and data handling (data pre-processing and post-processing). The model building and data handling phases are accomplished under the control of the Executive, which executes as a task under the host operating system.

The model building phase provides for the creation of the STMs, Scenarios, and SSMs. An SSM is created by issuing the following functional commands (in either the batch or time-shared environment):

- Select a TM (e.g., Airloads on Airfoils).

- Select, include, or delete processing options (e.g., select a table lookup process instead of a computation).

- Name and save the newly defined STM.

- Create and save a narrative description of the STM.

The System creates an STM control table that defines the configuration, logical flow, and any processing options selected by the user for the STM. This control information is saved in a System library. The user-supplied narrative is saved and linked to the STM. The creation of an SSM is accomplished in a similar fashion:

- Create a scenario by defining the processing logic flow for the model (e.g., process the Airloads module, then Downwash Module, then Blade Response Module).

- Define points at which checkpoint/restart records are to be produced, if desired.

- Name the SSM.

- Create and save a narrative description of the SSM.

- Assign files for external input data.

Based on the user's inputs, the System creates a control routine for the model, collects the software modules that collectively define the selected STMs along with their control tables, retrieves required System routines, initiates a job to compile all source routines and links all of the model components into an executable load module, and generates the job control language necessary to execute the model.

The creation of a Scenario consists of the principal activities performed by the user in creating an SSM as described above. A Scenario is the user-defined processing flow among STMs for an SSM. A Specific Scenario (SS) is a Scenario that is named and stored in the System library.

The model execution phase is initiated by submitting a job in the batch environment that initiates the model (SSM), initiating the model (SSM) as a task in the time-sharing environment, or by using the System to initiate the model (SSM). When an SSM is executed, all simulation data output is written to an intermediate System file for subsequent processing by the data handling phase.

The data handling phase of the System may be run as a job step subsequent to model execution or as an independent process. The data handling phase provides functions for printing or plotting model output data in System-defined standard output formats and provides functions for the creation of user-defined output formats. All System outputs are defined in terms of output groups and are associated with the individual TMs from which they are produced. Associated with each output group is a standard System output template and, for commonly generated output groups, one or more optional output templates. These templates define the format and contain other required information for the print and plot functions of the data handling phase.

C. HOW THE SYSTEM WILL BE USED

There are six basic steps involved in the use of the SAI/BV CHAS by the helicopter engineer:

- Definition of the engineering problem and review of the capabilities of the existing SSMs.
- Preparation of a new SSM (if required).
- Definition of input and output data sets.
- Execution of the SSM.
- Definition of output formats (if required).
- Processing of output.

Once the engineer has sufficiently defined his problem to be able to use the CHAS, he will review the collection of SSMs that exist in the CHAS library at his installation to see if the capability of one of them matches the definition of his problem. If an SSM exists in the library that exactly matches the problem that the engineer desires to solve, he may then proceed to defining the input and output data sets and then to the execution of the SSM. If, however, there is no such SSM, he must then either create an entirely new SSM or modify an existing SSM. The result of either procedure will be the

creation of a new SSM which then may be named and permanently stored in the library or temporarily stored for execution only at this time.

Once the SSM to be used exists (either chosen from the existing SSMs or created), the engineer must then define the input files that are to be used by the SSM and designate the variables which are to be output. This may involve the creation of new input files using the data handler. Once the input and output data sets have been completely defined, the SSM may be executed in one of three ways: interactively through the CHAS Executive, interactively through the time-sharing system of the host environment, or through the normal batch submission procedures of the host environment.

The final phase of using the CHAS is the obtaining of the desired output. This may involve either the use of existing System output templates or the creation of new output templates. When all of the output templates that are required have been prepared, the data handler would then be executed to process the output data and to provide it in the format specified by the various templates. After initial processing of the output, if further processing is desired, then new templates can be created and the data handler may again be executed to provide the additional forms of output.

D. <u>HOW THE SYSTEM WILL BE DEVELOPED</u>

The Second Generation Comprehensive Helicopter Analysis System will be developed over a 4-year period. To ensure that the resulting System complies with the requirements of the Government and industry users, it is essential that a thorough development plan be established for the development contract. This plan must define the following areas: organizational responsibility, development activities, documentation requirements, quality assurance requirements, configuration management, and testing requirements. The Development Plan devised by SAI/BV thoroughly addresses each of these areas.

The organization prescribed by the Development Plan ensures continuous involvement of all interested parties in the development of a CHAS. The organizations involved will be the Government Program Office (ATL), the Government-Industry Working Group (GIWG), the Technical Advisory Group (TAG), the Prime Development Contract (PDC), the Technology Integration Contractor (TIC), the First Level Technology Module Developers, and the Second Level Technology Module Developers. The interest of the Government will be represented by the Government Program Office and the TAG.

The GIWG will involve a wide range of Government agencies and industrial users. The PDC will be completely responsible for the development of the CHAS. In performing these duties, it will have the assistance of the TIC during the entire 4-year development activity. In order to ensure a diversity of industry participation in the evolution of the CHAS, it is proposed that the Technology Modules be developed by various helicopter manufacturers. During the First Level development, these Technology Module Developers would be subcontracted directly to the PDC. The Second Level Technology Module Developers would be either subcontracted to the PDC or contracted directly by ATL.

The organization of the System defined by SAI/BV is a hierarchical organization of four primary levels: System, subsystem, software segment, and programming. The top level is the System (CHAS). The subsystems of the CHAS are the Executive and the various Technology Modules. The software segments for the Executive have been defined during the predesign contract period . No software segments have been defined for any of the TMs during the predesign effort, but may be during a procurement of these TMs. However, a TM may not need to be further divided into software segments. Each software segment, or TM that has no software segments, is made up of programs. A program is the lowest level in the System for which any documentation will be provided. A Software Module is a technological rather than organizational grouping of computer code. An SM will consist of one or more programs and may or may not correspond to a software segment.

There are four primary phases in the development of the System: System design, subsystem design, software segment design and implementation, and integration. During each of the first three phases, essentially the same steps are accomplished.

During any of the three design phases, the requirements for the component (System, subsystem, or software segment) must first be defined. Each data item that is to be either input to the component, used by it, or output by it, must be defined and described. The data base supporting these requirements must also be described. At this point, the functions that are necessary to satisfy the defined requirements may be formulated. These activities result in the production of a Functional Description (FD), which is then submitted to a Functional Design Review (FDR) for approval. After approval of the FD, work can begin on defining the component in detail. This work will result in the development of a System Specification (SS), which would

then be submitted to a System Design Review (SDR) for approval. In the case of the System design and subsystem design, the decision activity is essentially complete at this point. Only the Test and Implementation Plan (PT) remains to be written. However, in the case of software segments, the next step is the development of the Program Specification (PS) for each program. This document defines precisely the logical flow within a program. The PS would be reviewed by ATL and the TAG to ensure compliance with the SS. After the PS has been completed, coding of the program can begin. After the program is coded, it will be subjected to unit testing. When all programs of a software segment have been unit tested, integration of the programs into the software segment can begin. Once integration has been completed, the PT for the software segment can be executed, and a Test Analysis Report (RT) written. Upon successful testing of the software segments that comprise the subsystem, the software segments can be integrated and the PT for the subsystem can be implemented. Again, an RT is written. After acceptance of the subsystem, they may in turn be integrated to form the complete CHAS. At this point, the total PT, which defines the acceptance testing, is put into execution. Successful testing will result in the acceptance of the System.

Schedule

The First Level System is scheduled for delivery at the end of two years. This release would be followed by three months of demonstration of the First Level Release. At the end of the 4-year period, the Second Level Release satisfying all of the requirements of the Type A system specification would be delivered.

Documentation

During the development of the CHAS, there will be three levels of documentation: System level, subsystem level, and software segment level. The authority for all documents produced during the development of the CHAS is MIL-STD-490 (Reference 1), or DoD Manual 4120.17M (Reference 2).

---

[1]Military Standard, MIL-STD-490, SPECIFICATION PRACTICES, Department of Defense, Washington, D. C., 30 October 1978.

[2]DoD Manual 4120.17M, AUTOMATED DATA SYSTEM DOCUMENTATION STANDARDS MANUAL, Department of Defense, Washington, D.C., December 1972.

The first document necessary at the System level is the Type A system specification, which has been produced as a result of the predesign effort. This document will form the basis for the procurement of the CHAS. The other documents required at the System level are defined in DoD Manual 4120.17M:

- Functional Description (FD),
- System/Subsystem Specification (SS),
- Data Requirements Analysis (RD),
- Data Base Specification (DS),
- User's Manual (UM),
- Computer Operation Manual (OM),
- Test and Implementation Plan (PT),
- Test Analysis Report (RT).

The same set of documents are also required at the subsystem level. A Type B5 computer program development specification, as defined in MIL-STD-490, is also required for a Technology Module that is to be procured separately.

The documents necessary for the software segments include those that are required for the System and subsystem levels. In addition to these manuals, a Program Specification must also be produced for each program. In the case of a TM that is not divided into software segments, PSs would be written at the TM level.

Each document, after approval, will be subjected to formal change control.

Quality assurance provisions are inherent in the development methodology. Some of the important features of those provisions are: The Functional Design Review, the System Design Review, the Change Control Review, Test Analysis Reviews, and the Code Reviews.

A configuration management plan will be produced as part of the responsibilities of the PDC. It requires that documentation become a configuration item after presentation at the scheduled review. A Change Control Board comprised of ATL, the PDC, and the TIC will review all engineering change proposals for any document that has been made a configuration item.

Formal testing will be conducted on all three levels: System, subsystem, and software segments. The respective Test and Implementation Plans developed at each level will precisely describe the tests to be conducted and how they are to be developed. The ultimate criteria for determining acceptance of the CHAS is its ability to produce a Specific Simulation Model, corresponding to a problem definition, which will execute on the host machine and which (a) correctly interprets all input types and value, accepting those that are legal and properly disposing of all others; (b) properly processes all internally stored data; (c) correctly formats, arranges, and outputs all required System data; and (d) is based upon valid engineering analysis.

The provisions of the development plan will ensure evolution of a CHAS that will provide the helicopter engineering community with an accurate solution at minimum cost.

1. INTRODUCTION

A predesign effort was conducted for development of the Second-Generation Comprehensive Helicopter Analysis System (CHAS). This effort was part of a program planned for the development and application of CHAS as shown in Figure 1.

This report summarizes the results of a predesign effort for the CHAS. A draft development specification (Reference 3) for CHAS was reviewed to determine the feasibility of development of such a system. The comprehensive requirements of CHAS include the requirements to analyze helicopter engineering problems involving:

- Performance
- Stability and control
- Loads
- Aeroelastic stability
- Acoustics

CHAS must have the capability to analyze a wide variety of helicopter configurations currently of interest to the potential Government/industry user community. A capability to represent the analysis problem at several levels of complexity for efficient use in different stages of the preliminary design/detailed design/research life-cycle phases must also be provided.

A preliminary design for the System was developed by SAI/BV based on the concept of a model builder/model user capability. This capability allows the engineer to draw from the technical capabilities developed for the general requirements of CHAS to "build" a computer program that meets his specific problem requirements without carrying along the burden of the general capability needed to meet the comprehensive requirements of CHAS. A model user could use the computer program without having detailed knowledge of the System. A

---

[3] BASELINE TYPE A SYSTEM SPECIFICATION, SECOND-GENERATION COMPREHENSIVE HELICOPTER ANALYSIS SYSTEM, Science Applications, Inc., and Boeing Vertol Company, 20 January 1978.

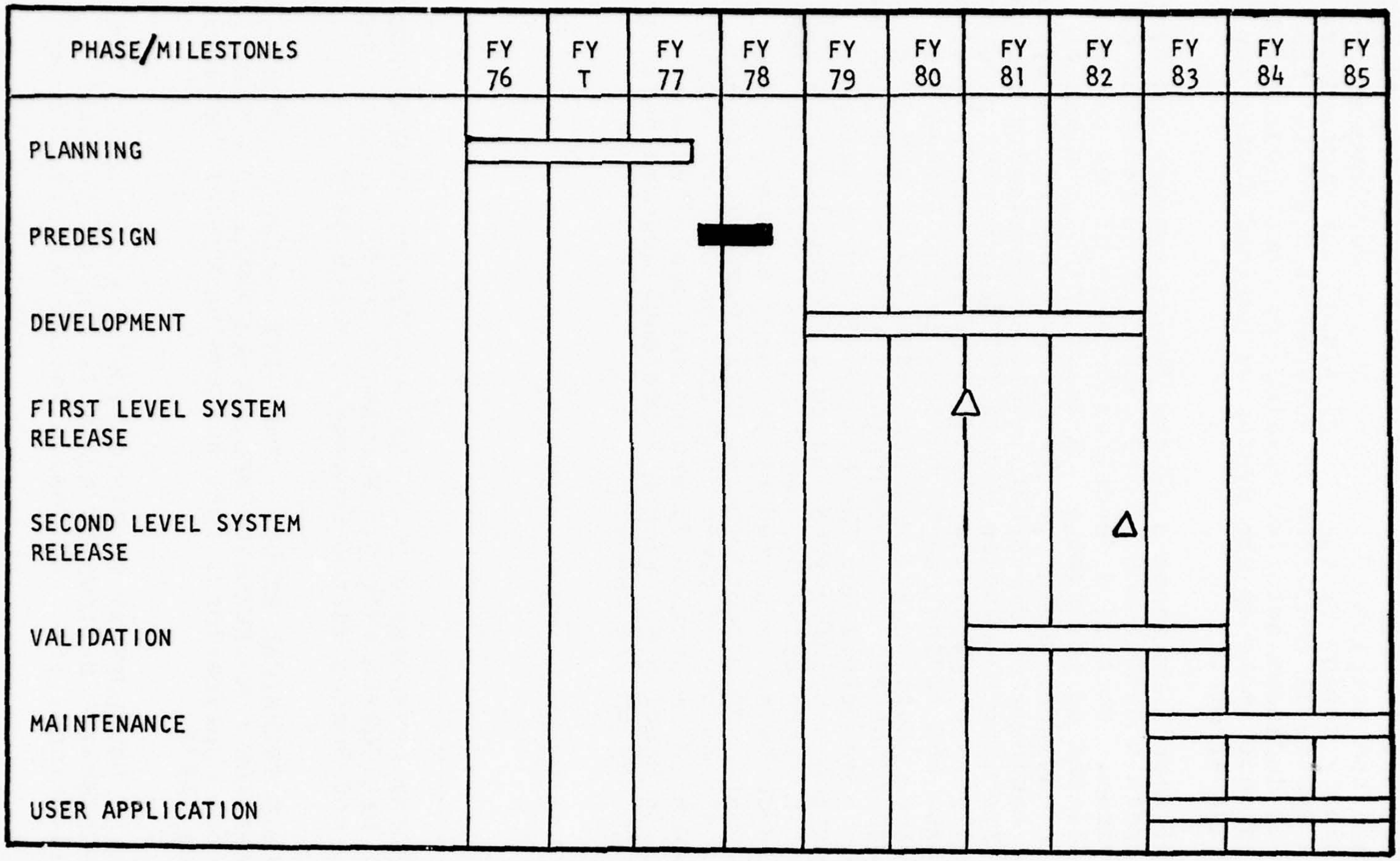


Figure 1. Overall Schedule - Second Generation Comprehensive Helicopter Analysis System

Baseline Development Plan was prepared (Reference 4) which outlined the details of the program for development of the preliminary design System. Analysis of the preliminary design System is presented in References 5 and 6. The analysis includes detailed development specifications for modules which make up the System as defined during the predesign effort.

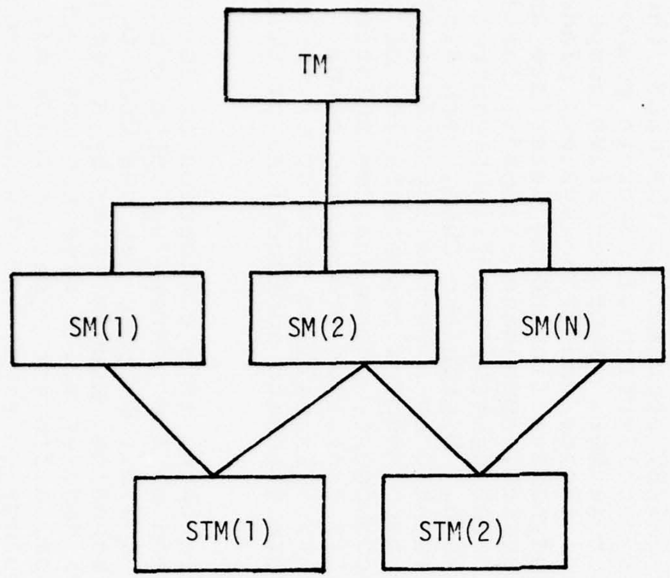
Section 2 presents a system design summary and Section 3 presents a discussion of the capabilities provided by the System. Section 4 describes details of use of the capabilities of the System. A description is given of how the engineer would use the System. Use of the System by the engineer at different levels of complexity including using Specific Simulation Models (SSMs)(computer programs for analysis of particular helicopter problems), developing new SSMs, and adding new technology to the System are discussed. Section 4.3 presents examples of use of a language (CHARLES) for developing Scenarios to define SSMs. Section 5 describes how the System will be developed, and Section 6 briefly puts the cost of developing the System in perspective by comparing its development cost with funds being requested by the Army and Navy for development and procurement of helicopters in fiscal 1979. A glossary of terms used throughout this report is presented in Section 7, and references cited are listed in Section 8. Figure 2 presents several definitions that are key to the CHAS.

---

[4] BASELINE DEVELOPMENT PLAN, SECOND-GENERATION COMPREHENSIVE HELICOPTER ANALYSIS SYSTEM, Science Applications, Inc. and Boeing Vertol Company, 7 March 1978.

[5] INTERIM TECHNICAL REPORT - TASK III ANALYSIS, SECOND-GENERATION COMPREHENSIVE HELICOPTER ANALYSIS SYSTEM, Science Applications, Inc. and Boeing Vertol Company, 22 May 1978.

[6] TYPE B5 DEVELOPMENT SPECIFICATIONS, SECOND-GENERATION COMPREHENSIVE HELICOPTER ANALYSIS SYSTEM, Science Applications, Inc. and Boeing Vertol Company, 22 May 1978.



TECHNOLOGY MODULE (TM): Contains the most complex representation of a particular technology including mutually exclusive processing paths.

SOFTWARE MODULE (SM): Sections of program code that represent the smallest subdivision of a complete technological representation of a helicopter component.

SPECIFIC TECHNOLOGY MODULE (STM): Represents a set of software modules chosen by the model builder applicable to the level of complexity of the subject engineering problem.

SPECIFIC SIMULATION MODEL (SSM): A simulation model (computer program), containing user-specified components (STMs), built by the model building system specifically for an engineering study.

SCENARIO: The defined flow among the STMs for an SSM.

SPECIFIC SCENARIO (SS): A named and stored Scenario that may be used as a building block in another Scenario.

Figure 2. Key Definitions

2. SYSTEM DESIGN OVERVIEW

2.1 What is the System? The SAI/BV concept of the Second Generation Comprehensive Helicopter Analysis System (CHAS) is that of a total system that will produce, upon user demand, a Specific Simulation Model (SSM) tailored precisely to the engineering problem under study. There are three phases to the overall system:

- The generation of a specific model
- The execution of a specific model
- Postprocessing of the output generated

The CHAS is designed to operate as a task under the host operating system with no changes required in the operating system for CHAS operation. The user will communicate all of his requirements to the CHAS through CHARLES, Comprehensive Helicopter Analysis Research Language for Engineering Studies. CHARLES is a single comprehensive language that facilitates all aspects of all phases of CHAS.

To facilitate the SAI/BV approach to the CHAS, the helicopter is viewed as a collection of discrete components that are coupled together. Representative components would be airloads, engine and drive system, blade dynamics, fuselage aerodynamics (including interface and fuselage airflow), acoustics, appendage airloads, fuselage dynamics, engineering aerodynamics, the flight control system, and rotor downwash. Within the CHAS, each discrete component will be represented by a Technology Module (TM), which will contain the most complex representation of the model of a particular component, phenomena, or solution method including mutually exclusive processing paths. In order to facilitate this, a TM will be made up of software modules that will perform discrete subfunctions of this representation.

A Technology Module then, is a collection of Software Modules and a description of the processing paths connecting the Software Modules. A Model Builder may use CHAS to construct a Specific Technology Module (STM) by selecting the particular Software Modules which give him the technology he desires. Thus, different Model Builders may select different technology to represent the same component of the aircraft.

The next step in building the SSM is to connect the various STMs together. This is done through the Scenario language of CHARLES. Once the Scenario has been formulated, the CHAS Executive can then be called on to produce a fully executable SSM. This model contains only the components that are under study by the user, and only the complexity for each of these components that the user desires.

Once the Model Builder has built his SSM, a user may execute it. This can be done in one of three ways. He may either execute it through the normal batch facility of the host computer, or he may execute it through the normal time-sharing facilities of the host computer. Alternatively, he may also enter the CHAS Executive and cause the execution of his SSM via the CHARLES language.

After the completion of the execution of a SSM that was built by the CHAS, the user then may initiate post-processing output of the results of that execution. This would be accomplished by entering the CHAS Executive and causing the output routines to produce his desired results. These results may be produced in either printed or graphic hard copy or on various display devices. Magnetic disc and tape files may also be prepared from the output for input to other SSMs or to models external to the CHAS.

In summary, the SAI/BV approach to the CHAS provides the helicopter engineer with the ability to easily construct a model that solves his specific problem, and to obtain the output of the results of this model in a variety of displays.

2.2 Major design considerations. The SAI/BV approach to the design of the CHAS was chosen to achieve three major objectives:

- An accurate solution
- Minimum cost
- Universal acceptance

In order to achieve an accurate solution, it is necessary that the Second Generation CHAS use proven state-of-the-art component technology. This technology

must be easily modified to include advances in the state of the art, and it must be capable of representing all the complex couplings of the helicopter. Achieving minimum cost will be accomplished by the use of only the technical capability and coupling complexity needed to solve the particular problem that the engineer presents to the System. The CHAS must also be able to efficiently manage available resources and to use minimum Executive overhead while executing. The actual model should also be machine interchangeable so that it does not have to be recreated in each user environment. The CHAS will be universally acceptable only if complete documentation is available and the System is completely validated. The CHAS must also meet all user community needs and be easy to use, in order to be universally acceptable. This acceptance can also be achieved through the shared development of various Technology Modules.

It is clear from the objectives that the System must be flexible and therefore contain a sophisticated executive that can manage data, provide for changes in levels of complexity and changes in couplings, and edit the technology to include new developments. In addition, the System must be efficient to minimize computation costs. The apparent conflict between flexibility (a sophisticated executive) and efficiency (computational costs) has been minimized by the SAI/BV approach to the CHAS Executive. This sophisticated Executive can provide all the needed flexibility and in addition be capable of building SSMs that exist independent of the building system, contain little (if any) Executive function overhead, and become load modules for execution of the analysis.

Consideration of these three main objectives has led to the formulation of the following technical requirements which must be met by the CHAS.

- The CHAS must satisfy all the analytical requirements as specified in Section 3.2.2 of the Type A system specification (Reference 2).

- It must be capable of consistently reducing the level of complexity of the representation of an engineering component, phenomenon, or solution method.

- It must provide coupling of components as desired to satisfy accuracy and cost constraints for a given problem.

In addition, the System Executive must have the following capabilities:

- A separate library of TMs and generalized mathematical routines that can be edited easily
- Evaluation of available storage so that the SSM may be constructed for the most efficient use of resources
- System generation of an SSM (load module) that can run interactively or in batch with a minimum Executive
- Generation of the load modules for various hardware systems
- Automatic documentation of SSMs
- User selection of output processing desired from available options
- User selection from a number of different input options

The System must be developed in such a manner as to provide the above capabilities and meet the three primary objectives. A development plan for the System must also contain the following features:

- Technology within the System must pass acceptance criteria, be correlated with other analyses and tests, and be completely validated.
- Technology Modules should be developed under separate contracts, awarded competively.
- First Level and Second Level systems must meet as many user needs as possible within available resources.

- Predetermined SSMs must be developed and be available to run in batch or interactive modes.

- The results of the predesign phase must be reviewed and the best concepts must be integrated into the System.

The concept for the CHAS formulated by the SAI/BV predesign effort satisfies all of these considerations. This system will provide both Government and industry users with a flexible tool that can grow to meet future and unanticipated requirements and developments in technology.

2.3 Mathematical basis.

2.3.1 Introduction. The overall concept of Technology Modules, a Model Builder function, and a Model User function has been introduced. This section reviews the mathematical considerations for this approach, which includes solution method and equation coupling.

The development of the SAI/BV mathematical approach is based upon satisfying the following requirements:

- The mathematical model required for a complete analysis of the helicopter is large and complex, and will grow larger and more complex in the future.

- Government and industry cost constraints require the engineer to efficiently utilize only the analytical capability needed to adequately solve his problem.

- To adequately analyze the different helicopter configurations (both existing and future), mathematical models with significant differences are required.

- The analysis is required to obtain accurate, cost-effective solutions, even though numerical methods algorithms are subject to wide variations in efficiency, reliability, and stability as a function of the mathematical model to be solved.

In order to satisfy these requirements, a System is defined that allows flexibility of both coupling and the solution method. This is accomplished by allowing the Model Builder to select the technology, the coupling, and the solution method suited to his problem. Examples of specific analyses built using this approach are given in the Appendix.

The discussion below reviews the understanding of solution methods and coupling attained during this predesign contract. The development contract would include a more detailed investigation, including consultation with experts in numerical methods.

2.3.2 Possible solution methods. There are two large classes of solution methods suitable for solving the helicopter mathematical model; these are:

- Undetermined coefficients
- Numerical integration of the initial value problem

The undetermined coefficients method would be used to obtain a steady state solution only. The method assumes that the forcing function is periodic, damping is positive, and the time since the initialization of the problem is so long that the transient solution of the differential equation has damped out. Therefore, only the steady state periodic solution remains. Under these conditions, the unknown variables can be expanded in an orthogonal periodic series of the same type as the forcing function (and its derivatives). Since all derivatives can be replaced by terms of the series, the only unknowns are the series coefficients, and the differential equations become algebraic equations.

This is ideal for analyzing the helicopter, since one rotor cycle makes a very convenient reference period. The forcing function (usually airloads) can easily be expanded in a Fourier series (based upon one rotor revolution). Expanding the unknown deflections in a Fourier series transforms the time domain differential equations into frequency domain linear algebraic equations that can be easily solved by inverting a matrix.

The numerical integration of the initial value problem approach approximates the solution of the differential equations by estimating the variable values at

discrete time intervals. The solution results as a time history of the variables. A large number of numerical techniques are available to perform these estimates (which can be very accurate). This method must be used when the assumptions of the undetermined coefficient method do not apply. This is true when transient, nonperiodic, or stability considerations are significant for the problem to be analyzed.

The initial value problem approach can be used for obtaining a steady state response by running the analysis until the transient response damps out. This approach for obtaining a steady state solution is not very efficient, and even though the efficiency can be improved by special techniques (like temporarily increasing damping) it is usually much less efficient than the undetermined coefficient method.

The best approach (minimized cost) for solving a nonperiodic problem is to use a combination of the two approaches. Choose the undetermined coefficient method to obtain the trimmed steady state response. Use the steady state response to define the initial conditions for the transient response problem. Solve the transient response problem by selecting a numerical technique to evaluate the initial value problem, at discrete values of time.

2.3.3 Steady state solution. A steady state response calculation will be needed for almost every problem. This includes defining:

- The initial conditions needed to start an initial value problem

- The baseline deflections about which linear perturbations are performed as part of a stability analysis

- The steady state helicopter response

The only time a steady state analysis is not needed is when the initial conditions or baseline deflections are known or assumed. Clearly, the savings resulting from using the undetermined coefficient method for determining the steady state response is well worth the small additional effort needed to include this solution technique in the System.

To solve a linear problem with undetermined coefficients, the simultaneous coupled differential equations representing the helicopter can be transformed into simultaneous linear algebraic equations, and solved with all coupling included by inverting the equation coefficient matrix. However, a linear formulation is not the usual mathematical model used.

Usually, airloads (the major helicopter forcing function) and certain coupling* terms involve significant nonlinearities. To analyze a nonlinear problem using undetermined coefficients, define the problem as an approximate linear system of equations, which includes all the nonlinear terms in the forcing function. Solve the problem by iterating between the linear response and the nonlinear forcing function (evaluated for the last response values) until a compatible (repeatable) solution has been obtained. For problems with large nonlinearities, linearized damping terms should be extracted from the nonlinear terms (evaluated about the mean values) and included in the linear equations to aid convergence. Specific examples are given in the Appendix.

2.3.4 Initial value problem. The SAI/BV approach to solving initial value problems is to allow the Model Builder to select from a library of numerical techniques the method best suited to the given problem. The task of providing interchangeable methods is simplified since over 90 percent of the explicit numerical methods used for solving the initial value problem require the differential equations to be expressed in the form:

$$\dot{y} = f \text{ (everything else)}$$

In our approach, each Technology Module that contains dynamic equations will be capable of expressing these equations in the form y = f ( ).** The selected solution method (from TM 15) will decide when y values are to be calculated and what values of time and displacement will be used for the calculations. These

---

* Rotor/fuselage coupling due to inplane forces includes nonlinear shortening terms and their derivatives.

** Additional software module(s) may be included in a Technology Module to convert the differential equations into any form.

decisions will be controlled by the DO WHILE statement in the CHARLES model building language and by control indices set in the solution method's Specific Technology Module. The actual sequence through the Technology Modules is controlled by the CHARLES language. Specific examples are given in Section 4.3 and a general example of solving an initial value problem is provided in the Appendix.

The main reasons for developing an interchangeable numerical method approach is that:

- Specific methods, applied to specific problems, can be inefficient, inaccurate, or unstable.

- There are many sophisticated analyses that can be used, and new methods are being developed.

The engineer would like to use the lowest cost and most accurate and stable numerical method to solve his problem. The interchangeable approach gives the engineer the ability to experiment by changing methods, to obtain experience, and to learn which method is best for his problem. A brief discussion of some sophisticated numerical methods is given below.

To minimize computer costs, smart numerical methods are available that perform error checks and can adjust the analysis time step and change the method's order. In addition, different numerical methods are available for solving problems with regular (smooth) behavior (by using past history), and problems with irregular behavior (by using current values only).

When the equations to be solved contain irregularities part of the time (such as airfoil stall), but are regular most of the time, dual numerical methods can be used. For example, a solution method assuming regular behavior can be used until indicators show that the mathematical behavior is experiencing irregularities. At this point, a different (but more costly) method can be used that adequately accounts for the irregularities. When indicators show that the irregular behavior is no longer present, the first method can again be used.

Techniques are also available for breaking the problem apart by using partitioned methods. When some of the component equations are smooth and some

of the equations are irregular, the problem can be partitioned, and different numerical methods used for the different equation types. To ensure proper coupling, an interfacing routine must be provided in the numerical methods package to properly account for the different time step size in the different partitions. For example, a smart, regular/irregular combined method (which is relatively expensive) can be used to determine the blade response (including stall effects), while a very efficient regular method (which is relatively cheap) can be used to find the fuselage response. The interfacing routine performs extrapolations, interpolations and error checks to ensure compatibility between the partitions.

All these methods require some additional computation to make decisions; therefore, their use does not automatically reduce computation time. Experimentation with different methods for a particular problem would be needed to obtain the most efficient, accurate, and stable solution. However, if efficiency is not critical (if the problem will not be run frequently or is very small), the experimentation need not be done, and a typical, middle-of-the-road (good accuracy and reasonable cost) method can be chosen, with a high probability of success.

2.3.5 Component mode coupling. Component mode synthesis (Reference 7) systematically combines linear components (in fixed or rotating coordinates) into a single system of fully coupled simultaneous linear equations. The eigenvalues and eigenvectors of this fully coupled system of equations can be found, and a subset of natural modes (including the effects of all components) can be selected to represent a system of equations with a reduced number of degrees of freedom. Therefore, component mode synthesis can conveniently couple independently defined components, and the resulting coupled system of equations can be simplified by selecting only modes of interest to represent the complete system (hence reducing the number of degrees of freedom and execution cost).

For example, consider a fuselage defined by 20 natural modes, and having no additional components. It

---

[7] Hurty, W. C., DYNAMIC ANALYSIS OF STRUCTURAL SYSTEMS USING COMPONENT MODES, AIAA Journal, Vol. 3, No. 4, pp. 678-684.

is desired to analyze an airframe with wings, engines, fuel cells, and absorbers attached to the fuselage. Assume that there are two of each component and each is represented by four modes. The component mode coupling will combine these nine components into a single set of 52 simultaneous equations, from which 52 eigenvalues and eigenvectors can be defined. All 52 modes can be used, or the problem can be simplified. To simplify the problem, the six modes closest to 4/rev can be selected to represent the fully coupled system with a four-bladed rotor. Therefore, in this example, the component mode coupling approach allowed nine separate fuselage components to be conveniently combined into 52 simultaneous equations, from which six degrees of freedom near the frequency of interest were selected, greatly simplifying the problem. This simplification applies to both the steady state problem and the initial value problem.

2.3.6 Coupling for the steady state analysis. Two methods are discussed below that can provide coupling for the steady state problem. Both methods are compatible with the SAI/BV system design, and both utilize component mode synthesis. The first method formulates and utilizes the fully coupled system equations, while the second method utilizes mobilities to break the problem into smaller parts. Though the methods are similar, it currently appears that the second method can offer significant cost savings for most problems. Further study would be performed as part of the development contract, and the possibility of providing both methods has not been ruled out.

The first method treats linear coupling by formulating the fully coupled equations of motion (for the complete system) using component mode synthesis (discussed in Section 2.3.5) and then solves these equations (or an equivalent orthogonal set) using the undetermined coefficient method. The modes of each component will be determined by the appropriate Technology Modules, and all the resulting modes will be coupled by the component mode coupler (TM 13B). The resulting fully coupled matrix equation can be either solved directly, or the eigenvalues/eigenvectors of the system can be determined. If the new eigenvalues/eigenvectors are used, the simultaneous equations can be replaced by an equivalent set of fully coupled orthogonal modes, which can then be solved independently. If desired, this set of equations can be simplified by selecting a subset of

the coupled orthogonal modes to represent the problem (see Section 2.3.5).

The alternate approach uses mobilities to represent portions of the system, hence breaking the problem into relatively small parts that are coupled to the rotor. For this approach, the modes of each component will be determined by the appropriate Technology Modules, subsets of the modes will be coupled by the component mode coupler* (TM 13B), and mobilities can be calculated to represent each linear subset. Using this method, mobilities can replace the actual equations representing the fuselage, drive system, control system, dampers, etc. These mobilities are included in the rotor blade equations, and the resulting blade solution automatically includes coupling with the mobility-replaced components. Response details for the mobility-replaced components, if desired, can be calculated from the coupled modes used to obtain the mobilities. (Note that if these details are not required, there is no need to calculate them.)

The advantage of the mobility method is evident when solving nonlinear steady state problems. Nonlinearities like aerodynamics, inplane hub loads, accurate lag dampers, etc., require that the solution be obtained through iteration (See Section 2.3.3). In addition, downwash coupling can best be analyzed using iteration. For example: nonuniform downwash from the main rotor induces airloading on the tail rotor and the fuselage which results in vibratory hub motion that feeds back to the main rotor. Though this coupling can be included in the linear coupled equations by using influence coefficients, it would significantly increase the complexity.

When iteration is used to solve nonlinear problems or to account for downwash coupling, the mobility method is more efficient, since it generally solves a much smaller set of equations. Essentially, only the blade equations must be solved for each iteration, since the mobilities are fixed and the detailed response of the mobility replacement components need not be calculated

---

* For any subset, eigenvalues/eigenvectors can be determined, and any fraction of the resulting modes selected to represent this subsystem.

until the final solution is obtained. The full matrix approach (i.e., equations of motion for the System generalized coordinates) must either repeatedly solve the fully coupled equations or calculate the eigenvalues/eigenvectors and replace the fully coupled system with an equivalent uncoupled (orthogonal) system. In either case the method requires calculating the response of all the degrees of freedom. An advantage of the full matrix approach is that it will be more convenient for supporting the aeroelastic stability analysis (TM 28).

A deficiency with the mobility method is that it cannot account for mismatched blades or cross-coupling without iterating or defining a larger system of equations (i.e., approximating the full matrix method). The first deficiency is due to the inability to transfer mobilities from the fixed system to the rotating blade root when the blades are mismatched. This problem can be solved by either iterating or by solving for the response of all blades in the rotor together. The second difficulty occurs because only one rotor is solved for at a time. This problem can be solved by either iteration or by solving for the response of all rotors together. The requirement to iterate to a solution for these two special coupling cases can cause the mobility method to be less efficient than the full matrix method. However, this is true only for a completely linear problem. If the problem is nonlinear, both methods require iteration, and since iteration coupling adds little additional cost, the mobility method should be significantly cheaper to run than the full equation method.

2.3.7 Coupling for the initial value problem. Component coupling for the initial value problem is implied in the solution technique, as long as the equations of motion for each component include all the coupling terms. To understand this inherent coupling, a simple problem will be discussed below. Consider a typical single degree of freedom problem. The equation of motion can be written in the form:

$$M\ddot{q}(t) + C\dot{q}(t) + Kq(t) = F(q,t) \qquad (1)$$

To get this equation into the required form (see Section 2.3.4), reduce the highest order derivative to the first order by substituting dummy variables. For this example, define a new variable y(t) such that:

$$y(t) = \dot{q}(t) \tag{2}$$

then

$$\dot{y}(t) = \ddot{q}(t) \tag{3}$$

Substituting Equations (2) and (3) into Equation (1) and retaining Equation (2) to recover the original variable, q(t), Equation (3) can be replaced by:

$$\dot{y}(t) = \frac{F(q,t)}{M} - \frac{C}{M}\, y(t) - \frac{K}{M}\, q(t) = f(t,y,q) \tag{4}$$

$$\dot{q}(t) = y(t) \tag{5}$$

Therefore, the differential equation has been replaced by two coupled equations of the required form.

Next, a solution method is selected. The solution method defines an algorithm for estimating the variable at a future time based upon current and past variable values. For this simple example, a second order algorithm will be used.

The first approximation is given by:

$$X(t + \Delta t) = X(t) + \Delta t\ (\dot{X}(t,X)) \tag{6}$$

where the term $\dot{X}(t,X)$ is the slope of X evaluated at time t.

The second approximation is given by:

$$X(t + \Delta t) = X(t) + \tfrac{1}{2}\Delta t\ [\dot{X}(t,X) + \dot{X}(t + \Delta t, \bar{X}(t, \Delta t))] \tag{7}$$

this approximation uses a better estimated slope by averaging the slope at time t, with the estimated slope at time t + Δt.

Applying the solution algorithm to the problem at hand is relatively straightforward. Using Equation (6), the first approximation for y and q is defined as:

$$\bar{y}(t + \Delta t) = y(t) + \Delta t\ [F(t, y(t), q(t))] \tag{8}$$

$$\bar{q}(t + \Delta t) = q(t) + \Delta t\ [y(t)] \tag{9}$$

Everything on the right side is known. The terms q(t) and q(t) are the initial conditions (and $y(t)=\dot{q}(t)$), and the term F(t,y(t),q(t)) is given by Equation (4), which is also a function of the known initial conditions.

Using Equation (7), the second (and final) approximation for y and q is defined as:

$$y(t + \Delta t) = y(t) + \tfrac{1}{2}\Delta t\ [f(t, y(t), q(t)) + f(t + \Delta t, \bar{y}, \bar{q})] \tag{10}$$

$$q(t + \Delta t) = q(t) + \tfrac{1}{2}\Delta t\ [y(t) + \bar{y}(t + \Delta t)] \tag{11}$$

Again, everything on the right side is known. The bar terms ($\bar{y}$ and $\bar{q}$) are known from Equations (8) and (9), and $f(t + \Delta t, \bar{y}, \bar{q})$ is obtained from Equation (4).

To continue calculating the solution of Equation (1), the above steps will be repeated again with the values of q and $\dot{q}$ at $t + \Delta t$ becoming the new initial conditions.

The example shows that the $\dot{y}$ = f( ) equation defines the problem. If there were many degrees of freedom there would be a corresponding first derivative equation for each, and as long as each equation contains all the relevant coupling degrees of freedom (as unknowns), the equations are automatically coupled. Whenever the numerical solution algorithm requires the first derivative value, all the coordinate values needed to evaluate it will be known, either as initial conditions or as previously calculated approximations. In fact, there is no reason why the equations cannot be evaluated by different Technology Modules or by different combinations of Technology Modules. Specific examples are given in the Appendix.

2.4 Identifying the Executive functions. The functions of the CHAS Executive are subdivided into four classes. The first class encompasses all functions that

are used to define the composition, structure, and logic flow of a simulation model. This class of functions is referred to as model building functions and the person who normally utilizes these functions is referred to as the Model Builder. The second class of functions, classified as model execution functions, support the run time processing of a simulation model. The third class, data handling functions, support the preprocessing of model input data and postprocessing of output data generated by the simulation model. The last class, Executive support functions, includes all functions that support the model building and data handling classes of functions plus the library maintenance support function. The basis for the functional groupings described above are two primary design objectives:

- Minimize the Executive overhead associated with the simulation model at model execution time.
- Separate the data handling functions, input data preprocessing and output data postprocessing, from the model building and model execution functions to provide a functionally independent Executive component.

The following subgroupings of functions and the individual functions within each subgroup were identified and allocated to the model building class of Executive functions.

- Specific Technology Module Build Functions
  - Technology Module Selection
  - STM Configuration Selection
  - System Narrative Selection (TM Specific)
  - User Narrative Creation (STM Specific)
  - STM External Output Selection (General Output Groups)
  - STM Input Requirements Identification
  - Name and Catalog STM
  - STM Processing Option Selection
- Specific Simulation Model Build Functions
  - Scenario Definition (define overall model flow)
  - Specific Scenario Definition (define Scenario as distinct System entity)

- External Input File Assignment
- Intermediate Output Data Selection (PAUSE)
- Model Checkpoint Selection (for restart processing)
- Model Timer/Trace Selection
- Scenario Control Program Generation
- Model Documentation Generation
- Model Job Control Language Generation
- Name and Catalog SSM
- Initiate SSM Execution
- Transfer Source SSM to Transportable Media

The following functions were identified and allocated to the model execution class of Executive functions.

- Checkpoint Handler (creates checkpoint records when called by the Scenario Control Program)

- Timer/Trace Handler (provides optional model timing and software trace outputs for debugging and efficiency evaluation)

- Diagnostic Handler (provides a centralized error message handling facility)

- Output Handler (provides a single interface point for all (nonintermediate) model outputs)

- Model Initialization (inputs external common data, optionally lists input data)

- Model Termination (calls Output Handler to write output trailer records and close files, lists model run statistics)

The following functions were identified and allocated to the data handling class of Executive functions.

- Output Template Generator (defines external data formats for System-generated data)

- Output Data Format (reformats and outputs System-generated data in accordance with a specified output template)

- Input Template Generator (defines reformatting and data validation requirements for converting a user-specified data file into the required System input format)

- Input Data Format and Validate (reformats, edits, and outputs user-specified data files in System format in accordance with a specified input template)

- Pen Plotter Interface (interfaces with the host system pen plotter routines)

- Graphic CRT Interface (interfaces with the host system graphics routines)

- Nongraphic Device Plot (provides a plot capability on a line printer)

- Data Reduction and Statistical Function (provides for the reduction of gross model outputs according to user-specified criteria, provides gross output evaluation aids)

The following functions were identified and allocated to the Executive support class of Executive functions.

- Batch Handler (provides the command input interface in the batch environment)

- Terminal Handler (provides the user interface in the conversational environment)

- Phase I Subsystem Executive (controls all Executive functions for model building and data handling functions)

- Data Base Manager (handles all file input/output for model building and data handling functions)

- Compiler (translates CHARLES commands)

- Converter (converts data from one system of measure to another)

- Restart (provides restart processing for simulation models)

- Tutorial (provides a conversational mode teaching aid to guide users in the use of System functions)

- Library Maintenance (provides for the creation and maintenance of all static System libraries)

2.5 Identifying the technology functions. Technology Modules were allocated to the system to represent the physical components of the aircraft and to compute the technical characteristics of the aircraft based on the following criteria:

- A Technology Module was selected to represent a physical component (or a subsystem of related physical components), phenomenon, and solution methods.
- TMs were purposely made small to provide flexibility in development of the System.
- TMs entry and exit points were selected to require a minimum transfer of data between TMs

TMs generally fell into categories that:

- Represent physical component properties (including air mass)
- Couple component or subsystem equations
- Perform a logical operation (trim)
- Perform standard mathematical operations

TMs selected and the function they perform are described briefly in the next section.

The procedure used to identify TMs to satisfy the Type A system specification requirements (Reference 1) was:

- Compile Type A specification requirements:
  - From body of Sections 3 and 4 of Type A specification
  - From Tables 3.2.1, 3.2.2, and 3.2.3 of the Type A specification
  - From the partial lists of PFCs and DFCs provided by ATL
- Assign the compiled Type A specification requirements to requirements for individual TMs.

2.5.1 CPCI and Technology Modules Summary. TMs allocated to the System are given in Table 1. These TMs makeup the General Functional Capability of the technology of the analysis system and there is a CPCI (Computer Program Configuration Item) for each TM. (A CPCI is a portion of a computer system which is to be separately procured.) These definitions provide a brief introduction to the functions performed by each TM. TMs are described in more detail in the Type B5 development specifications (Reference 6). The definitions include notes that show that several TMs previously defined in the development of the System have been deleted or combined with other TMs. TMs have generally been given a numerical designation, e.g., TM04. In some instances a letter designation follows the number, e.g., TMO6B, TM06C. Modules with the same number designation but different letter designations generally perform the same function with a different level of complexity or perform related functions. However, they are included in the overview architecture of the TM for that technology.

2.6 The combined System. The combined CHAS is a system that will be continually evolving. The Technology Modules existing at a given time in the System life cycle represent a bounded set of potential simulation models which may be realized through the exercise of the model building functions of the Executive. The Development Plan for the First Level and Second Level CHAS provides for the implementation of a set of Particular Functional Capabilities (PFC) representing current state-of-the-art rotorcraft simulation technology. This set of PFCs will then provide the baseline to which new methods and technology, representing advances in the state of the art, may be added. In the system design approach developed by SAI/BV, the PFCs designated for development as the First Level System and subsequently for the Second Level System will be implemented as SSMs. In this particular instance, each PFC will be implemented as a single SSM with no processing path options. This distinction is made because the model building functions of the Executive provide the capability of configuring an SSM with multiple processing path options. Further analysis will be performed to determine if, in certain cases, it would be more desirable to implement multiple PFCs in a single SSM.

Table 1. Technology Module Summary
TM01 - ROTOR BLADE DATA (checks distributed blade properties data and converts into a discrete form). TM02 - ROTOR BLADE EIGEN SOLUTION (assembles data and calls module to compute blade natural frequencies and modes). TM03 - BLADE FORCED RESPONSE-MODAL (uses blade modal properties to compute blade steady state and transient response to blade loads and hub and control motions). TM04 - ROTOR HUB (computes six degrees of freedom of rotor hub and rotor shaft steady state and transient motions and loads; couples rotors and airframe). TM05A - ROTOR TRIM FOR STEADY HUB FORCES (determines rotor control inputs to obtain prescribed steady, zero harmonic, rotor hub forces and moments). TM05B - ROTOR TRIM FOR MANEUVERS (determines control settings to obtain prescribed maneuver conditions such as banked turns and specified g pullups). TM06A - SIMPLE ROTOR DOWNWASH (computes uniform downwash or downwash linearly varying with radius). TM06B - ROTOR DOWNWASH-RIGID WAKE (rigid wake geometry is determined using induced velocities based on momentum theory; the Biot-Savart law is applied to individual wake elements and induced velocities at the rotor disc are calculated). TM06C - ROTOR DOWNWASH-DEFORMABLE WAKE (an iterative procedure is followed to determine consistent wake geometry and induced velocity field). TM07A - AIRLOADS ON AIRFOILS (computes radial distribution of airloads, lift, drag, and pitching moment acting on the rotor blade; computes airloads on fuselage lifting surfaces).

Table 1. Technology Module Summary (Continued)

TM07B - CIRCULATION CONTROL AND REACTION DRIVE (adds effects of circulation control; computes and adds effects of reaction drive).

TM07C - UNSTEADY AERO OF FLAP (computes aerodynamic effects of a flap attached to an airfoil).

TM08A - ROTOR CONTROL SYSTEM-KINEMATIC (specifies blade control motion due to control system input for a rigid control system).

TM08B - ROTOR CONTROL SYSTEM-FLEXIBLE (accounts for flexibility in the rotor control system including flexibility of swashplates, actuators, control linkages, pitch links and arms).

TM09 - ROTOR INITIALIZATION (calculates rigid or flexible blade response estimate to start flexible blade rotor analysis).

TM10 - TRANSFORMATION (included in other TMs).

TM11A - RIGID FUSELAGE AND TRIM LOGIC (performs rigid fuselage force balance and trim logic for steady state trim; includes external cargo, on ground and towing or winch-down operations).

TM11B - FREE FLIGHT FUSELAGE TRIM (defines fuselage flight path as a function of control input).

TM11C - EXTERNAL LOAD AND SUSPENSION SYSTEM (defines coefficients of equations of motion for external load subsystems about the nominal trim position; determines component motions from system motions).

TM11D - PRESCRIBED MANEUVER-FUSELAGE (defines hub loads required to perform specified maneuvers).

TM11E - VARIABLE BOUNDARY FUSELAGE (defines fuselage "flight path" and contact loads for: landing, including fixed or moving deck; towing; winch-down; taxiing).

Table 1. Technology Module Summary (Continued)

TM12A - SIMPLE FUSELAGE AIRLOADS (uses table lookup for aerodynamic coefficients to calculate fuselage airloads).

TM12B - AIRLOADS FOR FUSELAGE STORES AND EXTERNAL CARGO (computes airloads on bluff bodies such as external loads and fuselage stores).

TM12C - FUSELAGE POTENTIAL FLOW (determines flow field and force and moments acting on a body of arbitrary shape).

TM13A - AIRFRAME INTERFACE (organizes data for coupling engine/drive system, fuselage, and external load subsystems to form an aircraft system - exclusive of rotors).

TM13B - GENERAL MODE COUPLER (couples components for a subsystem or couples subsystem to form a system using the method of component mode synthesis).

TM14 - FUSELAGE (determines coefficient matrices for the coupled fuselage subsystem equations of motion from uncoupled component equations of motion and component properties; determines component loads and motions from subsystem loads and motions).

TM15 - NUMERICAL METHODS LIBRARY (performs general matrix operations; computes eigenvalues, eigenvectors; performs numerical integration).

TM16 - NONLINEAR BLADE DAMPER (determines damper forces from relative blade and hub motions).

TM17 - NONLINEAR VIBRATIONS CONTROL DEVICES (computes vibration control device output vs blade, hub, or airframe input).

TM18 - ROTOR BLADE STRESS (computes rotor blade stresses from rotor blade forces and moments and rotor blade section properties).

Table 1. Technology Module Summary (Continued)

TM19 - HUB MOBILITY (combined with TM04 - ROTOR HUB).

TM20 - AUXILIARY PROPULSION (included in TM24-EXTERNAL FORCES; may be added later if a detailed, separate definition of auxiliary propulsion is required).

TM21 - ACOUSTICS PACKAGE (computes near field, far field, and internal noise spectra including rotor, engine, transmission, and weapon noise).

TM22 - GUST GENERATION (generates the effects of an aircraft penetrating a gust).

TM23A - AUTOMATIC CONTROL SYSTEM (senses aircraft motions and control inputs; modifies control inputs to improve flight characteristics; provides for a modular control system development. A control system may be developed for any requirement, e.g., vibration control system such as higher harmonic control).

TM23B - EXTERNAL LOAD STABILIZATION (senses aircraft and/or external load motions and creates forces between the fuselage and load suspension elements to reduce motions).

TM24 - EXTERNAL FORCES (specifies time history of external forces such as those due to gun firing and auxiliary propulsion; specifies where forces are applied and magnitudes and directions of forces).

TM25 - AIRCRAFT WEIGHT, INERTIA, GEOMETRY (uses component and/or subsystem weight, inertia, and geometry data to compute total aircraft weight, c.g. location, and rigid-body inertias - without rotors).

TM26 - ENGINE/DRIVE SYSTEM (compiles engine/drive system component stiffness, weight, inertia, gear ratio, geometry, and constraint data for assembly into a subsystem).

Table 1. Technology Module Summary (Continued)

TM27 - ENGINE SPEED-FUEL CONTROL (senses engine speed and adjusts fuel flow to maintain engine speed within prescribed limits).

TM28 - AEROELASTIC STABILITY - (CONSTANT COEFFICIENTS - determines hover air resonance and ground resonance stability characteristics for helicopters with symmetric rotors; three or more blades per rotor with equal azimuthal spacing. PERIODIC COEFFICIENTS - perturbs generalized coordinates of the system to obtain the Floquet transition matrix; determines aeroelastic stability mode damping from the eigenvalues of the transition matrix. TIME HISTORY - determines aeroelastic stability mode damping from decay data).

TM29 - CHANGE COMPONENTS (defines changes to component data due to movement or dropping a component or due to failure or damage).

TM30 - STABILITY AND CONTROL INDICATORS (determines stability derivatives and uses a low frequency representation of the aircraft/rotor system equations of motion to determine aircraft flying qualities).

TM31 - MOVABLE COMPONENTS (combined with TM29-CHANGE COMPONENTS).

TM32 - ENGINE PERFORMANCE (computes engine performance parameters).

TM33 - DYNAMIC AIRFOIL FLAP (defines structural dynamic characteristics of a flap attached to an airfoil).

TM34 - TIME SERIES ANALYSIS (performs fast Fourier transform and auto- and cross-correlation data analysis).

3. SYSTEM CAPABILITY

3.1 What the System can do. The capabilities of the System will be delivered in two releases: First Level and Second Level. The technology capability of the First Level Release will provide the minimum technology required to represent a helicopter. This limitation was necessary due to the resources available and the need to concentrate on the Executive capabilities during the first 2 years of the project.

3.1.1 First Level capability

3.1.1.1 Executive. The development schedule described in detail in Section 5.2 requires that most Executive capabilities be included in the First Level Release. These capabilities fall into seven main areas: model building, language processing, library maintenance, subsystem control, terminal interface, data handler, and environment.

The model building capability consists of the following major functions:

- Allow the user to specify the functions and flow he desires for the representation of a specific helicopter technology and validate the logic of the user-defined flow with the Technology Module (TM).

- Compile and build the Specific Simulation Model (SSM) using the Specific Technology Modules (STMs) built; add the routines for timer/restart capabilities and cause the executable SSM to be developed.

- Accept from the user his desired flow for the simulation model, which implies the coupling of the various TMs; generate a control program that couples the STMs together.

- Accomplish all interface functions associated with user directives when the CHAS is operated in the batch mode.

- Provide for tracing the flow, the time, and the resources of each STM as the SSM executes.

- Build the files that allow the SSM to be interrupted and restarted.

- Allow for conversion of one system of units to another.

Language processing consists of the following major functions:

- Interpret the user commands and produce a processing code and the data, including syntax and lexical analysis.

- Produce, based upon a code imbedded in the software, the textual tutorial stored in a System library.

- Produce, based upon a code imbedded in the software, the diagnostic message stored in the System library.

Library maintenance has the primary function of allowing the user to create, enhance, and maintain all files required to operate the CHAS. The files that will be created and maintained through the library maintenance system are:

Specific Technology Modules
Software Modules
Common Routines (Math Packs)
Specific Simulation Models
Specific Scenarios
Diagnostics
Tutorials
Data Files
CHAS Modules
Coupling Tables
Validation Files

The library maintenance program will interface to these files through the CHAS data base manager or the operating system access methods as appropriate.

Subsystem control provides internal control of the System during the model building phase of CHAS. Requested commands presented by other segments are passed to the appropriate segments providing the required function.

The terminal interface accomplishes all interface functions associated with the interactive user. The terminal interface communicates with the host terminal I/O package and with the functional subsystems required by the user.

The data handling phase of the System, based upon user directions, produces the output in presentable form from the raw data produced during the SSM execution or provided as input data.

The environment consists primarily of the data base manager program which is the interface between the System modules that require input or output and the operating system programs that drive the actual storage devices.

3.1.1.2 First Level Release technology capability. The First Level System Release will provide capability to analyze the following technical characteristics:

- Performance
- Loads
- Stability and Control
- Aeroelastic Stability (time history analysis).

Technology Modules that will be developed for the First Level Release are shown in Table 2.

Rotor blade distributed properties data will be used to define a lumped parameter blade model with TM01. A finite element (NASTRAN-type) model of the rotor blade is proposed, with centrifugal stiffening effects added. TM02 will be used to compute rotor blade natural modes and frequencies (eigenvalues/eigenvectors). A modal representation of the rotor blade would then be used to compute blade response to air loads and hub motions. Summing of the effects of all rotor blades would occur in TM04. TM05A would be used to determine rotor trim conditions for a prescribed set of steady hub forces, steady flight conditions, and aircraft attitudes and motions. TM06A would be a simple rotor downwash model; TM07A would compute airloads for both nonrotating and rotating airfoil sections. TM08A would provide a kinematic representation of the control system, and TM09 would be used to initialize the rotor calculations.

Table 2. Technology Modules to be Developed for First Level System Release

CPCI ID	FUNCTION
TM01	ROTOR BLADE DATA
TM02	ROTOR BLADE EIGEN SOLUTION
TM03	BLADE FORCE RESPONSE - MODAL
TM04	ROTOR HUB
TM05A	TRIM FOR STEADY HUB FORCES
TM06A	SIMPLE ROTOR DOWNWASH
TM07A	AIRLOADS ON AIRFOILS
TM08A	CONTROL SYSTEM-KINEMATIC
TM09	ROTOR INITIALIZATION
TM11A	RIGID FUSELAGE TRIM LOGIC
TM11B	FUSELAGE MANEUVER (ARBITRARY CONTROL INPUTS)
TM12A	SIMPLE FUSELAGE AIRLOADS
TM13A	AIRFRAME INTERFACE
TM14	FUSELAGE
TM15	NUMERICAL METHODS LIBRARY
TM30	STABILITY AND CONTROL INDICATORS
TM34	TIME SERIES ANALYSIS

TM11A would provide trim logic for the rigid fuselage while TM11B would be used to control a time history analysis of aircraft response to control inputs. TM12A would be used to compute airloads on the fuselage (exclusive of rotors and nonrotating airfoil sections attached to the fuselage). TM13A provides the capability for coupling the drive system to the airframe and the rotor systems; although not needed at First Level Release, provisions must be made for incorporating this module into Second Level Release.

TM14, Fuselage, uses fuselage component data to define fuselage properties; component weight data are organized for later computation of aircraft weight, c.g., and inertia data. Equations of motion for the fuselage are developed and fuselage geometry is defined (rotor locations, aerodynamic surface locations, etc.).

TM15, Numerical Methods Library, is a collection of standard numerical procedures plus any special numerical procedures required to perform operations such as:

- Matrix Operations
- Eigenvalue/Eigenvector Solutions
- Numerical Integration
- Convergence Test Procedures (for Trim Solutions)

A separate module, TM34, Time Series Analysis, has been identified for analysis of time history decay data; this module might be combined with the collection in TM15.

TM30 will be used to control the determination of aircraft stability and control indicators. Included will be determination of stability derivatives and setup for eigenvalue and eigenvector computations.

A capability for rotor performance computation will be inherent in the procedures for airloads computation for the individual blades and for the summing of effects of all blades on a rotor. Total aircraft performance will be determined with effects of the influence of individual components. Simplified SSMs will be available for rapid computation of performance parameters. A capability for loads and vibration computation will be provided, which will include:

- Rotor blade loads and motions
- Airframe vibration (motions may be externally substituted into a NASTRAN-type finite element analysis to obtain airframe internal loads)
- Overall control system loads (with a total control system stiffness included in the analysis).

The capability for control system analysis will include determining aircraft stability derivatives, corresponding eigenvalues and eigenvectors, and time to half amplitude. A capability to determine time histories for response to control inputs will also be provided.

Aeroelastic stability analysis will be performed by the time history method where damping data are computed from decay data following cyclic excitation of the aircraft/rotor system. This approach provides the capability to include effects of nonlinearities.

3.1.2 Second Level Release capabilities.

3.1.2.1 Executive. As described in Section 3.1.1.1, most Executive capabilities will be provided with the First Level Release. However, the Second Level Systems will have the following additional capabilities:

- Enhanced SSM execution efficiency
- Transportability of SSMs
- Enhanced tutorials and diagnostics

3.1.2.2 Second Level Release technical capability. The Second Level System Release will add the capability for computing acoustic characteristics (TM21) and the capability for computation of aeroelastic stability with constant coefficient and periodic coefficient representations (TM28). TMs to be added between the First Level and Second Level Release are shown in Table 3.

TM5B will provide the capability to determine rotor trim and control inputs to achieve prescribed maneuvers. Representation of the control system will be expanded to include flexible effects in both the rotating and fixed system (TM8B).

Table 3. Additional Technology Modules to be Developed for Second Level System Release

CPCI ID	FUNCTION
TM5B	ROTOR TRIM FOR MANEUVERS
TM6B	ROTOR NONUNIFORM DOWNWASH (RIGID WAKE)
TM6C	ROTOR NONUNIFORM DOWNWASH (DEFORMABLE WAKE)
TM7B	CIRCULATION CONTROL AND REACTION DRIVE AIRLOADS
TM7C	UNSTEADY AERODYNAMICS OF A FLAP
TM8B	MECHANICAL ROTOR CONTROL SYSTEM (FLEXIBLE)
TM11C	EXTERNAL LOAD
TM11E	VARIABLE BOUNDARY FUSELAGE
TM12B	AIRLOADS FOR FUSELAGE STORES, ETC.
TM12C	FUSELAGE POTENTIAL FLOW ANALYSIS
TM13B	GENERAL MODE COUPLER
TM16	NONLINEAR BLADE DAMPER
TM17	NONLINEAR VIBRATION CONTROL DEVICES
TM18	ROTOR BLADE STRESS CALCULATIONS
TM21	ACOUSTICS PACKAGE
TM22	GUST GENERATION
TM23A	AUTOMATIC CONTROL SYSTEM
TM23B	EXTERNAL LOAD STABILIZATION
TM24	EXTERNAL FORCES
TM25	AIRCRAFT WEIGHT/INERTIA
TM26	ENGINE/DRIVE SYSTEM
TM27	ENGINE SPEED - FUEL CONTROL
TM28	AEROELASTIC STABILITY (TIME HISTORY ANALYSIS IN LEVEL 1)
TM29	CHANGE PHYSICAL PROPERTIES
TM32	ENGINE PERFORMANCE
TM33	DYNAMIC AIRFOIL FLAP

Major component representations to be added include TM11C, External Load, and TM 26, Engine/Drive System. Additional component representation capability will be provided in TM11E, Variable Boundary Fuselage; TM13B, General Mode Coupler, TM16, Nonlinear Blade Damper, TM17, Nonlinear Vibration Control Devices; TM25, Aircraft Weight/Inertia; TM29, Change Physical Properties; and TM33, Dynamic Airfoil Flap. TM13B provides a powerful capability for coupling component modes of subsystems of the aircraft. TM11E provides a capability to impose motions on the aircraft (such as ship deck motion) and to compute the approach to and contact with a boundary such as in landing on a moving surface of a ship.

Additional aerodynamic analysis capability will include: TM6B and TM6C, Rigid and Deformable Wake Downwash representations; TM7B, Circulation Control and Reaction Drive Airloads; and TM7C, Unsteady Aerodynamics of a Flap. TM12B will add the capability for computing airloads on such items as fuselage stores while TM12C will add a capability for potential flow analysis of an arbitrarily shaped body.

In the control system area, TM23A will provide a modular control system representation so that any automatic flight control system may be developed from a standard set of control system elements and any new elements may be added by developing a new software module. An external load stabilization system capability will be added with TM23B; this module might draw on the general capability of TM23A. TM27 will provide engine speed/fuel control capability and again might draw on the general capability of TM23A.

TM32 will add engine performance computation capability. TM24 will provide the capability to apply arbitrary external forces to arbitrary points on the aircraft (for auxiliary propulsion, gun firing, etc.), and TM18 will provide a capability to compute rotor blade stresses from rotor blade loads and rotor blade section properties. TM13B, General Mode Coupler, provides an additional loads computation capability for components; TM13B is based on Hurty's component mode synthesis method (Reference 7). It provides a capability for determining reactions between components and component mode generalized coordinate data for back substitution of motions into a component finite element representation (extended to CHAS) for determining component internal loads.

3.2 <u>How the System can be used</u>. The CHAS offers the user several options in terms of System access, model definition (model building), model execution, and data preprocessing or postprocessing (data handling).

At the highest level, the user has the choice of batch or interactive access to the System. This choice may be nullified for a specific installation due to the host processor's hardware/software configuration or by local operating procedures, but the System is designed to operate in either mode depending upon the specific operating environment. Certain applications, such as model execution requiring extremely large memory resources, or models that run for a long period of time, may be impractical in the interactive (time-shared) environment. However, the model building and data handling functions (input data preprocessing, output data postprocessing) are ideally suited to the interactive environment.

Independent of the user's choice of System access methods, the user is given an option in selecting a simulation model that satisfies his current analysis requirements. The First Level and Second Level Releases of the CHAS will be delivered with a number of Government-selected Particular Functional Capabilities (PFC). "Each PFC is representative of a type of analysis task frequently encountered by the user community, ..." (Reference 3). If the user's modeling requirement is satisfied by one of the System-supplied PFCs, the user need only to develop the necessary input data and proceed to the model execution phase. The steps in using an existing SSM are outlined in Figure 3.

If, however, the user's modeling requirement differs from the System-supplied PFCs or if he simply wishes to develop a very simple component representation before utilizing a more complex representation as implemented in a System PFC, then he may employ the Executive model building functions to develop an SSM that is tailored to his specific requirements. (Alternatively, the user may need a more complex model than that supplied as the System PFC, in which case the System-supplied PFC could provide the basis for the more complex model.) The steps in building new SSMs and adding new technology are outlined in Figure 4.

- <u>SIMPLEST CASE: USE EXISTING SSMs</u>
  - USER DEFINES HIS PROBLEM
  - USER REVIEWS CAPABILITIES OF AVAILABLE SSMs
  - USER SELECTS SPECIFIC SIMULATION MODEL (SSM) WHICH MATCHES HIS PROBLEM REQUIREMENTS
  - USER REVIEWS SSM INPUT REQUIREMENTS DEFINED

    IN SSM DOCUMENTATION
  - USER DEFINES HIS INPUT DATA SETS (FROM CARDS, TAPE, AND/OR PERMANENT FILES)
  - USER SUBMITS JOB FOR EXECUTION
  - USER DEFINES OUTPUT PROCESSING FROM OPTIONS AVAILABLE FROM SSM AND HOST SYSTEM CAPABILITIES
  - OUTPUT PROCESSING IS ACCOMPLISHED

- <u>PAUSE</u>
  - PERMITS INTERRUPT TO REVIEW INTERMEDIATE RESULTS AND THEN CONTINUATION OR TERMINATION OF EXECUTION

Figure 3. How to Use the System with Existing Specific Simulation Models (SSMs)

- BUILDING NEW MODELS (SSMs) AND ADDING NEW TECHNOLOGY

  - MODEL BUILDER

    - DEVELOPS NEW SCENARIOS AND GENERATES NEW SSMs

    - EDITS SCENARIOS

    - USES A SPECIFIC SCENARIO FROM LIBRARY IN COMBINATION WITH STMs TO GENERATE NEW SCENARIOS AND NEW SSMs

    - GENERATES NEW STMs FROM EXISTING TMs to USE IN NEW SCENARIOS; NEW SCENARIOS ARE USED TO GENERATE NEW SSMs

  - TO ADD NEW TECHNOLOGY

    - DEFINE TECHNOLOGY FOR NEW SOFTWARE MODULE (SM) IN AN EXISTING TM

    - OR DEFINE SMs FOR A NEW TM

    - CHECK CONSISTENT INTERFACE WITH THE SYSTEM

    - MODIFY LOGIC TABLES FOR VALID PATHS THROUGH TMs (i.e., VALID STMs)

    - MODIFY SOFTWARE MODULE DESCRIPTIONS, INPUT REQUIREMENT DEFINITIONS, LIBRARY OF VARIABLE NAMES, ESTIMATES OF RESOURCE REQ'Ts, ETC.

    - HAVE NEW SMs, TMs CODED AND ENTERED INTO THE LIBRARY SYSTEM

Figure 4. How to Build New Models (New SSMs) and Add New Technology

In the instance where the user elects to develop a new model for his particular application, the initial step could be a review of the current CHAS library listing. The CHAS library listing is analogous to a system master parts list composed of sets (SSMs), assemblies (SSs), subassemblies (STMs), and parts (the individual software modules constituting TMs). From this listing, the user may determine at which component level he will begin to assemble his simulation model.

For example, if the engineering problem required an initial simple trim computation, the user would look for an assembly (represented by an SS) that provided this function. The search for such a function is facilitated by the organization of the library listing into major component groupings and by generic classification within major groupings. Therefore, the user need only look under the SS grouping for the Trim generic classification to determine if an SS exists that meets his needs.

If no trim scenario is found, the user may then check the next lower component grouping, Specific Technology Modules. From this grouping, the user may pick existing STMs which, in combination, provide the simple trim processing the user desires. A hypothetical example of the desired trim processing might include the following STMs: (1) Aircraft Weight and Inertia, (2) Rotor Initialization, (3) Fuselage Airloads, (4) Airloads on Airfoils (Fuselage), and (5) Rigid Fuselage Trim. Assuming that each of the desired STMs existed in the library, the user could proceed to create his SS for trim processing.

Now assume that the STMs existing in the library did not quite fit the user's processing requirements. Perhaps the STMs for Airloads did not include a nearwake calculation. The user may utilize the STM build function to access the Airloads STM and create a new Airloads STM that includes a nearwake calculation. In defining the new Airloads STM, the user may elect to incorporate the near wake calculation as a processing option. In this way the user may defer, until model execution, the decision to execute the nearwake calculation in the trim processing.

Up to this point, several options have been described on how to use the System. The preceding has described some of the possible steps and some of the System functions that may be utilized in preparing for the execution of an SSM. To summarize to this point, the user has a choice of System access modes (batch/interactive). The user may utilize a System-supplied PFC (SSM) or he may use a combination of System-supplied/user-defined software components that range from SS to STM in order to define an SSM that is tailored to the unique requirements of a particular analysis problem. If the tailored SSM approach is taken and a new SSM is created in the process, the user may incorporate processing options into the new STM. A more detailed description of functions available to the user, e.g., defining check points for a model or defining a conditional processing path in a SS, is provided in Section 4. The next logical step in how the System can be used deals with the execution of the model, the SSM execution phase, and the data handling function that formats the model outputs and also preprocesses the model input data if the user so chooses.

Once the user has identified the SSM that meets his needs, he may wish to verify certain input data prior to model execution. The Executive data handling function provides functions for range checking and plotting System input data sets and also provides functions for reformatting data sets into the format expected by the System.

The Executive maintains input and output templates for all TMs. These templates define the format for all possible System inputs and outputs and also specify valid input data value ranges where applicable. The data handling function provides a facility whereby the user may define an input template that describes the format of an input data set which is not in the standard System format. Using a function of the Executive data handler, the user may direct the System to reformat the nonstandard input data set into the standard System format. This function could have application in instances where it is desirable to use data generated by an external model or where data was developed for another system and the format differs from the required.

Once any desired data preprocessing operations have been successfully concluded, the user has several alternative methods of initiating execution of the SSM. One alternative is the submission of the job in accordance with local system procedures for a batch run. This would include the submission of a valid JOB card and any additional job accounting control cards required by the host system. These control cards would be followed by a CALL or EXEC or equivalent control card to invoke the cataloged procedure that was produced by the Executive when the SSM was created. The only additional control inputs would be input file assignments to override the System default assignments. The specific form of these assignments would vary according to the host operating system and will be described in the machine-dependent supplement to the CHAS User's Manual.

Another alternative for SSM initiation is via the CHAS Executive itself. This form of model initiation totally divorces the user from all operating system considerations. By initiating the CHAS Executive in the interactive environment, the user may invoke an SSM in the interactive mode or the model execution job may be entered into the system batch processing stream. Alternately, the user may initiate the CHAS in the batch environment and, using CHAS commands, direct the Executive to submit the desired SSM to the batch queue and also supply, in the form of CHAS language directives, any input file overrides that may be necessary. Figure 5 shows schematically the model building interaction with the Executive, the Library resources, and its resulting SSM and its documentation.

The final use of the System concerns the extraction and presentation of a model's output in a form that facilitates the engineer's interpretation of the results. The data handling functions of the Executive provide for plotting and printing of SSM outputs in either the batch or interactive environment. By utilizing the data handling functions in the interactive environment, the user could review the SSM outputs at a gross level (e.g., plot or request a tabular listing of critical data values using a large scale factor or a specific value range qualifier) in order to select certain output data subsets for subsequent output in the batch environment.

Figure 6 shows schematically the use of an SSM and postprocessing of the results produced.

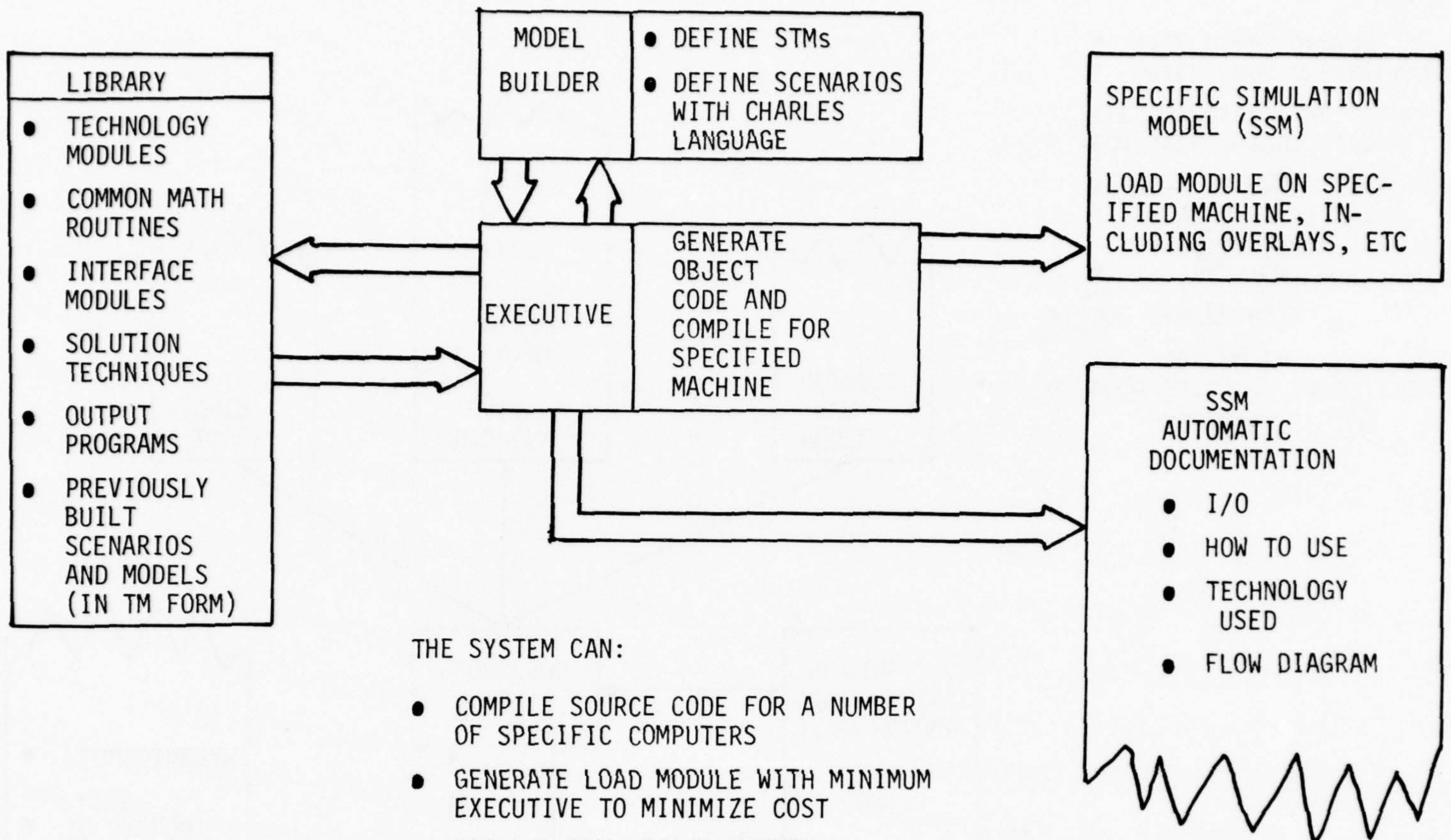
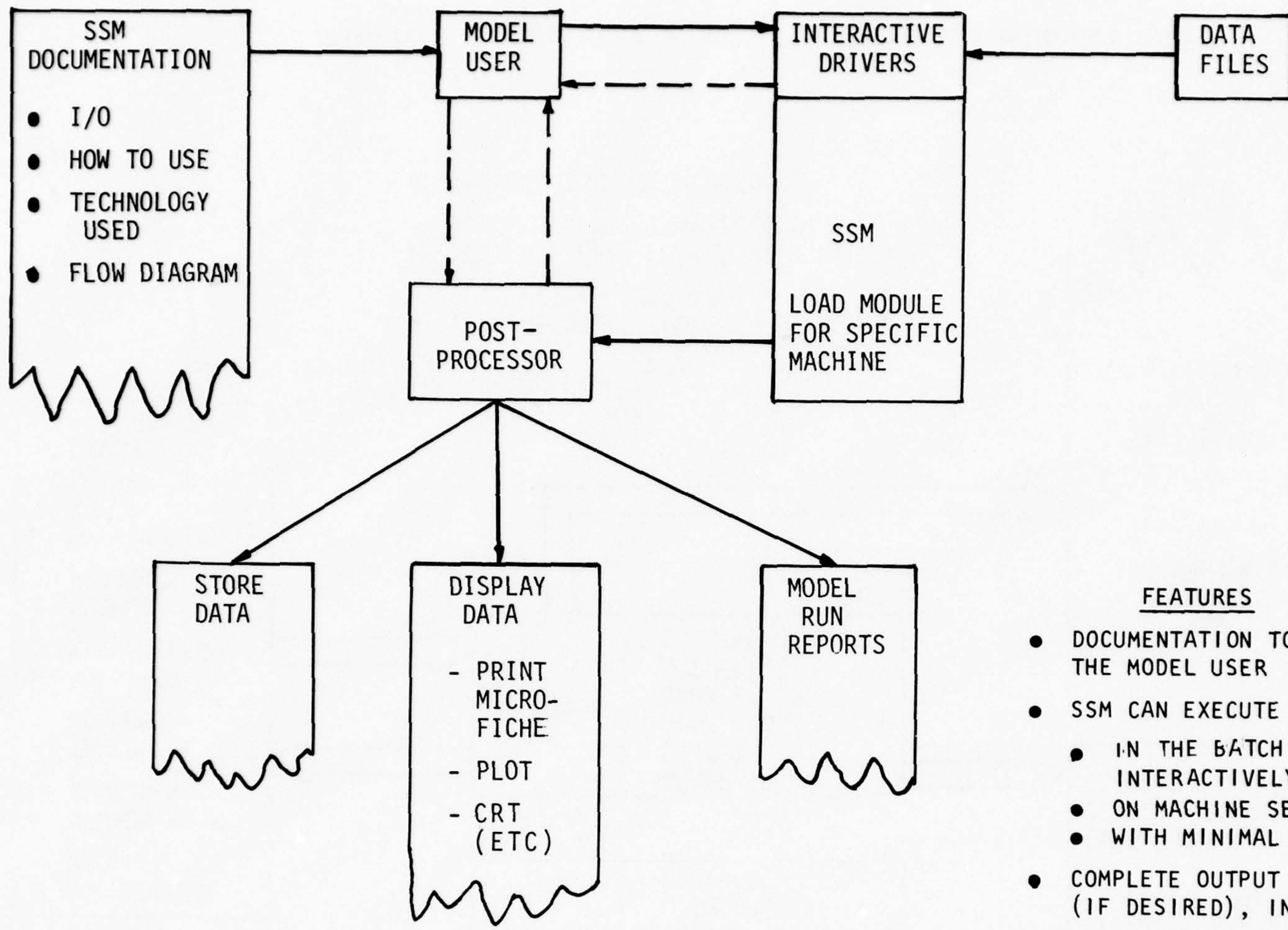


Figure 5. **Building a Specific Simulation Model (SSM)**



FEATURES

- DOCUMENTATION TO ORIENT THE MODEL USER
- SSM CAN EXECUTE
  - IN THE BATCH MODE OR INTERACTIVELY
  - ON MACHINE SELECTED
  - WITH MINIMAL 'EXECUTIVE'
- COMPLETE OUTPUT CONTROL (IF DESIRED), INCLUDING:
  - QUICK LOOK CAPABILITY
  - DATA STORE OPTION

Figure 6. Use of a Specific Simulation Model (SSM)

4. HOW TO USE THE SYSTEM

4.1 The Model Building Executive. The preceding sections presented an overview of the CHAS and described the general System capabilities. Section 3.2 described how, in terms of general functional capabilities, the System can be used to develop, execute, and format output from a simulation model representing a helicopter engineering problem. This section describes the CHAS in terms of specific Executive functions of:

(1) Model Building
includes the creation of Specific Technology Modules (STM) from Technology Modules (TM) and the creation of Specific Simulation Models (SSM), which includes the development of Specific Scenarios (SS) and the final step, creation of a stand-alone executable load module, the SSM, including the job control language (JCL) necessary to run the model.

(2) Data Handling
includes the preprocessing (printing, plotting, value range checking) of model input data for verification purposes and postprocessing (data reduction, printing, plotting, reformatting) of model output data to aid the user in the evaluation of model outputs and also to provide a mechanism for restructuring CHAS model outputs into a form acceptable to an external model.

This section also contains descriptions of the System diagnostic process, the automatic documentation facility, and the library maintenance function.

4.1.1 Preparation for model building. Prior to using the System for the actual creation of a simulation model in response to engineering requirements, the model building user can query the System for information concerning data resident in the System library. First, he can examine all TMs to determine the most complex representations of helicopter technology that are available. He can examine all previously built STMs to determine their applicability to his problem. Likewise, he can study previously built Specific Scenarios (SS) and other SSMs. Armed with the data concerning

the availability of building blocks constructed by other model builders, he can proceed with the development of his model, the first step of which is the definition of STMs.

4.1.2 Building Specific Technology Modules. The primary purpose of the STM concept is to allow the Model Builder to select a TM level of complexity that is in consonance with the engineering problem under study. To accomplish this the user requests the STM BUILD function. He then requests the technology desired, e.g., AIRLOADS. The template for the overall architecture for the AIRLOADS TM is then presented to the user. A sample is presented in Figure 7. From this template, the user can create the STM required for his problem.

To obtain an STM there are several possibilities. One possibility is to use decision points to define options.

For example, there are five decision points (T1-T5) defined in the template. To build an STM, assume the user defined the following decisions:

T-1 go to 4
T-2 go to 6
T-3 go to T-4

The language will allow the user to simply choose these options. Only the decisions defined in the template can be made in the STM building phase. The System will validate the user's selections.

Figure 8 depicts the flow of the STM defined by the choices made above. This STM is then available for model building where a simple AIRLOADS representation is requested.

As the System evolves, new concepts can be added to a TM to enhance that component technology within the CHAS. Using the same example of AIRLOADS, assume new techniques were available and the TM developer wished to modify the AIRLOADS on AIRFOILS (TM-07) to include:

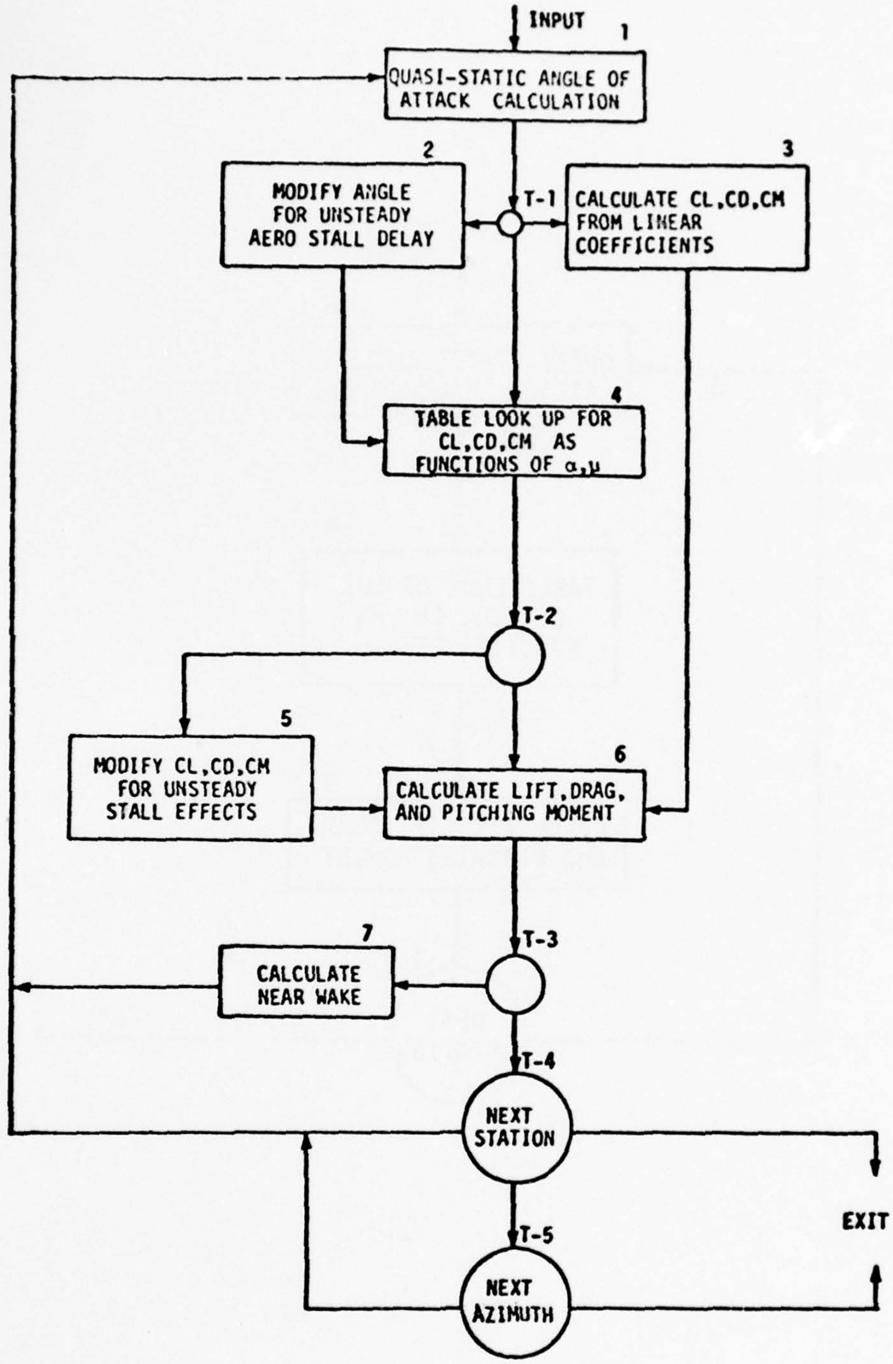


Figure 7. Software Modules of a Technology Module (Airloads)

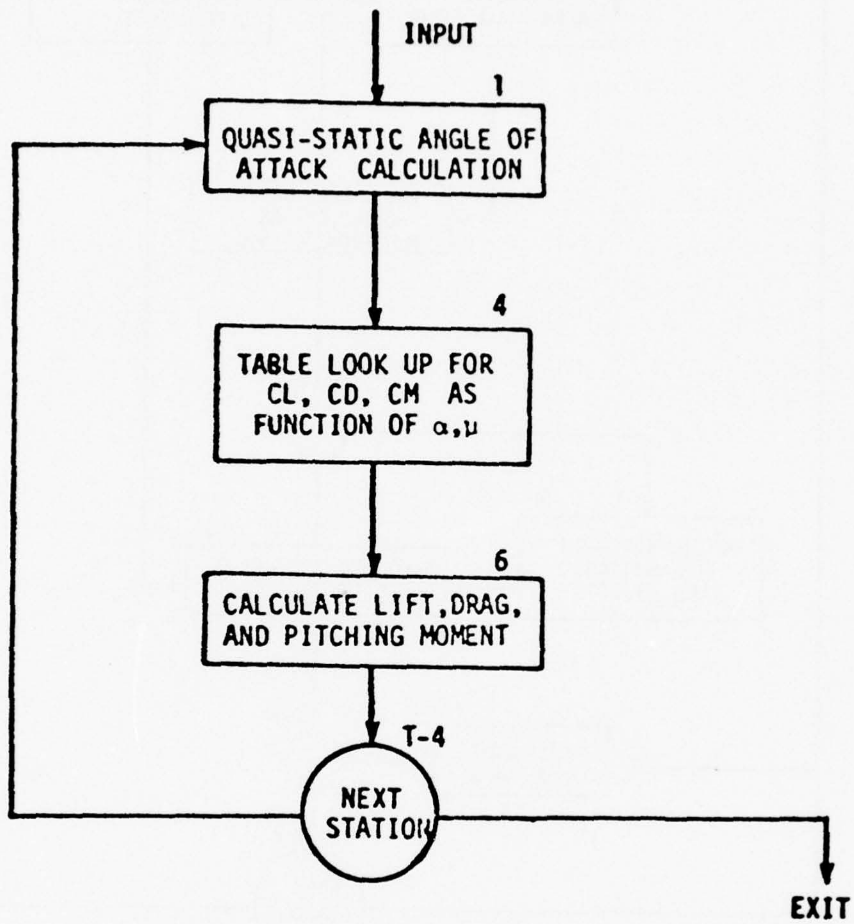


Figure 8. Specific Technology Module for Airloads

Circulation Control
Boundary Layer Control
Airfoil Flap

This would be done by defining a new template, with software modules 8, 9, 10, 11 and tests T-6 and T-7 included in the TM. This new template, shown in Figure 9, would then be available to Model Builders for STM creation. The STMs become the building blocks for the next phase of model building, SSM BUILD.

4.1.3 Model building, Specific Simulation Models. The final step, and primary objective, of the model building capabilities of the System is the definition of an SSM. The Model Builder accomplishes this by defining a Scenario, the flow among the STMs, and the conditions for this flow. This specification is done in the scenario language of CHARLES, which contains the following statement types:

(1) Specific Scenario (SS) Declaration. (This means execute the stated Specific Scenario.)

(2) Specific Technology Module Declaration. (This means execute the stated STM.)

(3) DO WHILE

(4) IF, THEN, ELSE

(5) Assignment, e.g., X=1

(6) PAUSE

(7) CONTINUE.

This language allows the Model Builder to specify the flow of the SSM. To further define these concepts, an example has been defined.

Figure 10 depicts the logic flow for an SS that will determine trim. In this SS, called TRIM1, STM TM25/1 (which was created from TM25) is executed. The variable T is tested for a condition of (<1) and if true, four other STMs are executed in a serial mode. SS names may be assigned any meaningful symbol string in accordance with the standard practices of a particular installation. Synonyms may also be allowed.

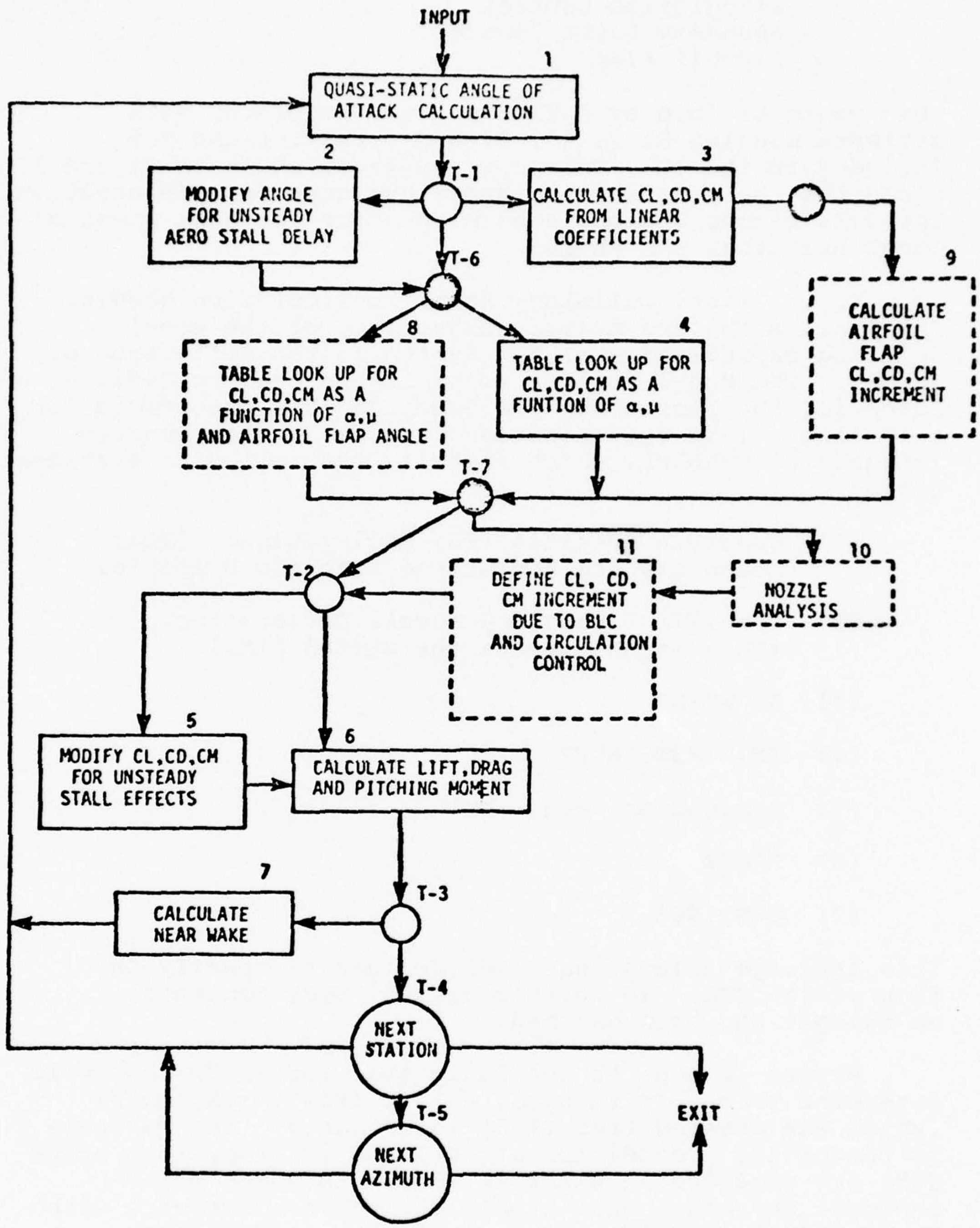


Figure 9. Technology Modules with Additional Software Modules

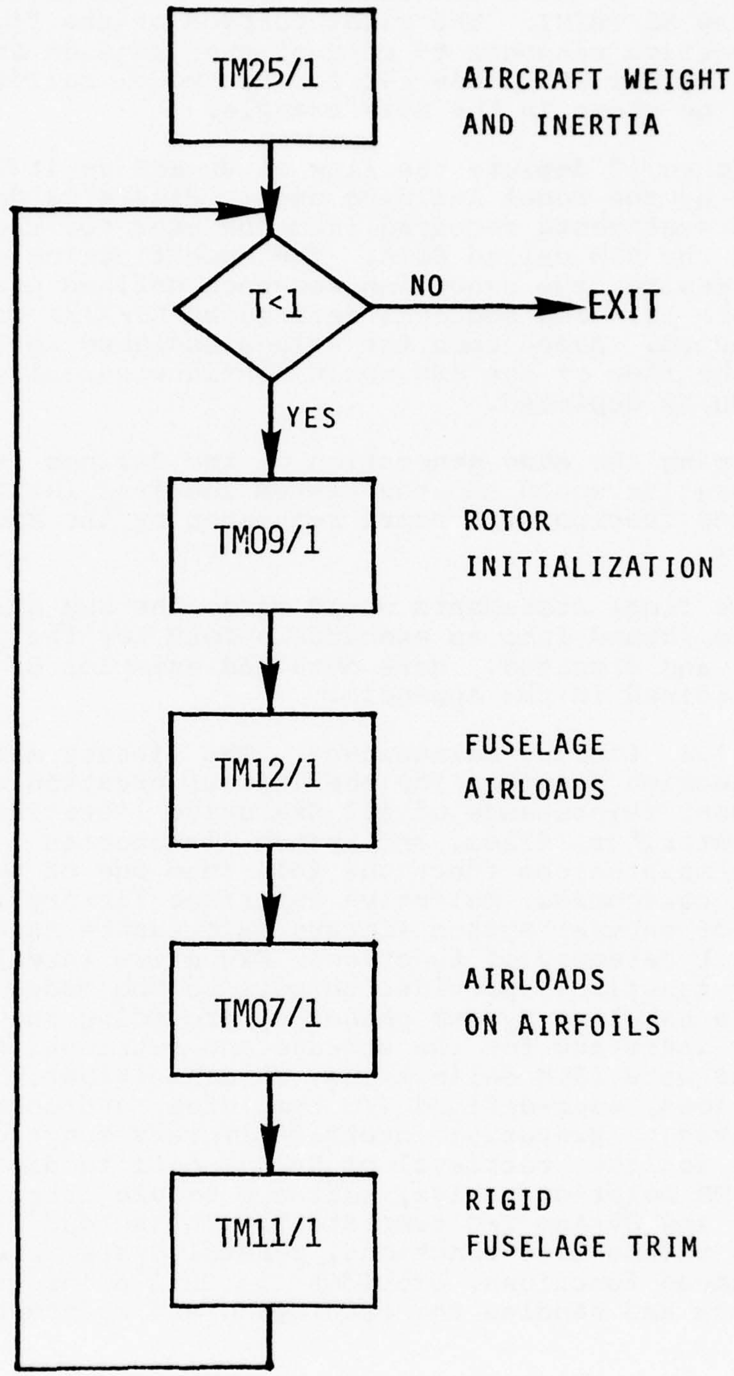


Figure 10. SS TRIM1 Logic Flow

Figure 11 shows the CHARLES statements required to define SS/TRIM1. The right portion of the figure shows the Executive response to each of the language statements. This SS is now available for further model building functions as will be shown in the next example.

Figure 12 depicts the flow of an SSM as it would be defined by the model building user. Figure 13 depicts the CHARLES statements required from the user for the development of the SSM called SSTA. The specification of the SS/TRIM1 requires the execution sequence defined previously in Figure 10. The sequence defined by SS/RI/1 would then be executed. Based upon the values assigned to X within STMs, the flow of the SSM would continue serially through the STMs as depicted.

During the code generation of the defined scenario, the Executive would add the system routines for Timer/Trace and Checkpoint/Restart requested by the Model Builder.

The final statements would cause the SSM SSTA to be saved, bound into an executable form for the target system, and executed. More detailed examples of SSMs are contained in the Appendix.

4.1.4 Library maintenance. The library maintenance function provides for the initial creation and subsequent maintenance of all Executive libraries, System attribute files, and System directories. All library maintenance functions fall into one of two general categories, Executive interface library functions, or general System library maintenance functions. The first category of functions, Executive interface library functions, provides support to the model building and data handling System phases by providing the System library interface for the storage and retrieval of user-supplied data (STM definitions, SS definitions, SSM definitions, user-defined I/O templates, and user-defined narratives). Executive interface library functions also provide for the retrieval of System-defined data (tutorial files, TM solution tables, software module attribute tables, and System I/O templates). The second category of library maintenance functions, general System library maintenance functions, provides the CHAS maintenance interface and handles the cataloging and maintenance of

USER COMMANDS	EXECUTIVE REPONSE
DEFINE SS/TRIM1	Initiate scenario build function, SS name=TRIM1 verify unique SS name.
TM25/1	Verify STM existence
DO 100 WHILE T.LT.1	Define T as a control variable issue warning message - flag T as uninitialized
WARNING-CONTROL VARIABLE T NOT INITIALIZED	Response displayed
TM09/1	Verify STM existence, check control variables
TM12/1	Verify STM existence, check control variables
TM07/1	Verify STM existence, check control variables
100 TM11/1	Verify STM existence, check control variables if loop contains no reference to control variable T, issue fatal diagnostic
END, SAVE	Resolve STM interface data requirements, note and list external SS data required. Save and catalog SS as 'TRIM1'
SAVED - TRIM1, 780502-10:50. EXTERNAL SS DATA REQUIRED: TM25/1 ARRAY - CWGTS SCALAR - C32	Response displayed

Figure 11. CHARLES Statements for Specific Scenario Definition

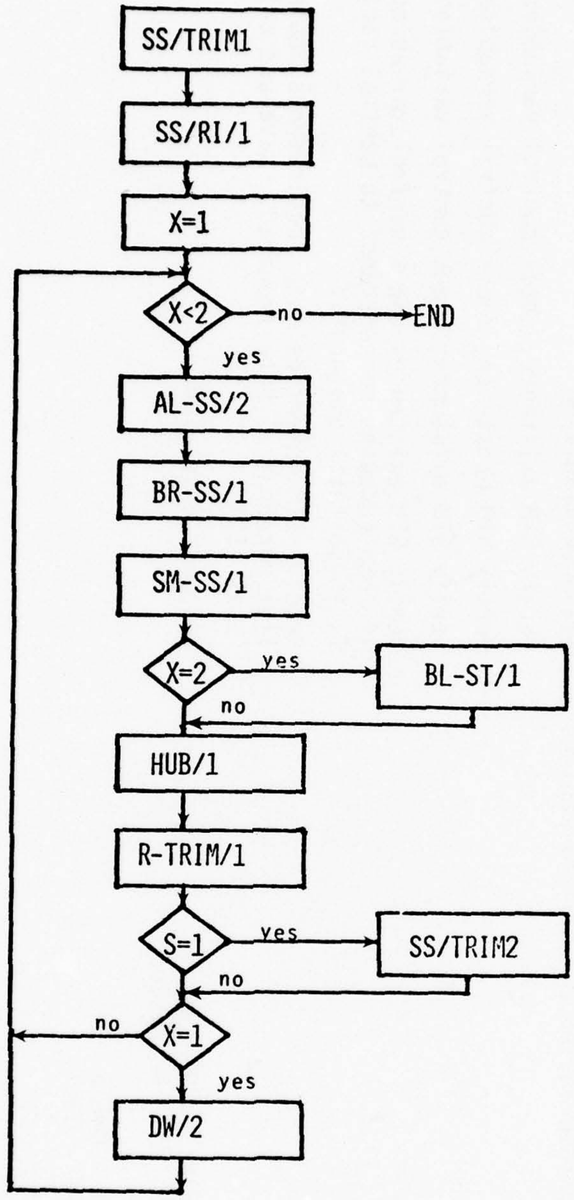


Figure 12. Flow of a Specific Simulation Model

USER COMMANDS	EXECUTIVE RESPONSE
DEFINE SSM SSTA	INITIATE SCENARIO BUILD FUNCTION SET SSM FLAG - SSM NAME = SSTA VERIFY UNIQUE SSM NAME
DEFINE SS RSS1	SS NAME = RSS1, VERIFY UNIQUENESS
SS/TRIM1	VERIFY SS EXISTENCE
SS/RI/1	VERIFY SS EXISTENCE
X = 1	DEFINE X AS LOCAL CONTROL VARIABLE
DO 110 WHILE X.LT.2	DEFINE LOOP BOUNDARY, FORWARD REF TO LABEL 110, FLAG X AS INITIALIZED
AL-SS/2	VERIFY EXISTENCE OF STM AL-SS/2
BR-SS/1	VERIFY EXISTENCE OF STM BR-SS/1
SM-SS/1	VERIFY EXISTENCE OF STM SM-SS/1
IF (X.EQ.2) THEN BL-ST/1 ELSE CONTINUE	VERIFY EXISTENCE OF STM BL-ST/1 X PREVIOUSLY DEFINED
HUB/1	VERIFY EXISTENCE OF STM HUB/1
R-TRIM1	VERIFY EXISTENCE OF STM R-TRIM1
IF (S.EQ.1) THEN SS/TRIM2 ELSE CONTINUE	DEFINE S AS CONTROL VARIABLE VERIFY EXISTENCE OF SS TRIM2
IF (X.EQ.1) THEN DW/2 ELSE CONTINUE	VERIFY EXISTENCE OF STM DW/2 X PREVIOUSLY DEFINED
110 CONTINUE	DEFINE LOOP END, REDEFINE X AS GLOBAL CONTROL VARIABLE (NOT SET IN LOOP). IF NO REFERENCE TO X WITHIN LOOP BOUNDS, ISSUE FATAL DIAGNOSTIC. IF NO REFERENCE TO CONTROL VARIABLE S IN CURRENT SCENARIO ISSUE WARNING DIAGNOSTIC
END RSS1, SAVE	NOTE AND LIST EXTERNAL SS DATA REQUIRED SAVE AND CATALOG SS AS 'RSS1'

LISTING OF EXTERNAL SS REQUIRED DATA

END SSTA, SAVE, BIND, EXECUTE

Figure 13. CHARLES Statements for SSM SSTA

system resource data (TM definition data, including SMs, solution tables, attribute tables, and I/O templates, plus all System tutorial files). The library maintenance function will utilize host processor library maintenance utilities, e.g., IBM IEBUPDT or CDC MODIFY, for the storage and maintenance of System software module source data. The identification, status, and attributes of all System resources will be maintained in CHAS directories and associated files. Library maintenance provides a library list function that produces tabular directory listings of user-defined library files and System resource library files.

4.1.5 Diagnostics. The diagnostics available to the users of the CHAS are at two levels: those for the model building user which are output by the System during SSM building or STM building, and those which are output to the engineering user during SSM execution. Before describing these two levels of diagnostics, the overall philosophy of storing and retrieving diagnostic messages will be described since it applies to both levels. SAI/BV have defined a concept of centralized diagnostics. In this concept, full diagnostic text messages are stored in a centralized file that is under the control of the CHAS library. These diagnostics may be added to, changed, or deleted, using the library maintenance function, independent of model building or model execution. In order to have a diagnostic message displayed to the user, the TM developer will embed in his program a code that is keyed to a record descriptor in the diagnostics file. Thus, when an error condition is encountered during System execution, the code can be displayed to the user, or, at his option, a full textual message can be displayed. This concept allows the diagnostics the user actually sees to be as long as necessary to explain the exact error condition that has just occurred. As the System evolves through use, these diagnostics can then be tailored by model builders and model maintainers to be most meaningful, based on previous experience by users who have encountered the same error.

4.1.5.1 Model building diagnostics. Model building diagnostics will be developed by the prime development contractor during the implementation of the

executive portion of the CHAS. These diagnostics will be designed to aid the user in the maintenance of the libraries, which contain Technology Modules and system programs, and in the development of Specific Technology Modules and Specific Simulation Models. In general, the diagnostics will be designed to keep the model building user from developing a Specific Technology Module, or a Specific Simulation Model, which contains an illogical flow. An illogical flow in this context means a flow through a Technology Module that was not conceived by the Technology Module developer. Additional errors that occur in processing due to System or hardware errors will also be output to the user. The difference between tutorials and diagnostics should be noted at this point. The centralized concept described in the preceding paragraphs for diagnostics applies also to tutorials. The primary difference between diagnostics and tutorials is that tutorials are presented to the user based on his request for help in determining what function is available next in the model building system. Additional tutorials are available to show him what his logical choices are. That is, the System will help the model building user to step through the process of building STMs and SSMs. These tutorials can be excluded upon command of the model building user after he has attained a level of expertise in the use of the System where he does not need the tutorial help. The combination of tutorials and diagnostics will allow the model builder to build STMs, build SSMs, and to maintain the libraries.

4.1.5.2 Model execution diagnostics. Diagnostics are available to the engineer using an SSM based upon the diagnostic codes embedded in the software modules by the TM developers. The concept of centralized diagnostics also applies to an SSM. That is, codes are embedded that may be presented to the user, or the user may choose a full textual description of the diagnosed error he has encountered. The full textual description will be available in one of two ways. If the SSM is executing on the host system of the model building portion of the Executive, the centralized library from the model building files will be available to an SSM during its execution for the extraction of diagnostic information. If the SSM is executing on a host system that does not contain the model building portion of the Executive, the System will build a file for the SSM from the library that contains the

diagnostics applicable to his model. During model execution, if an error condition exists within a software module, the code will be accessed and presented to the user at an interactive terminal. If the user is employing batch mode, the diagnostic will be sent to the system output device. Note that there are three levels of diagnostics: those which cause a fatal error, those which are a warning condition, and informative. Warning and informative diagnostics are presented to the user as described above. Fatal diagnostics are presented the same way, but also include an end-of-run generated by the System. This means that when the System issues a fatal diagnostic, it cannot be overridden by the user. The determination of what is a fatal diagnostic is contained in the error code. Therefore, it is the responsibility of the TM developer to determine the severity level, warning or fatal, for all diagnostics. This concept of diagnostics and tutorials allows the System to present to the user, in two forms, any type of error encountered during execution. It also allows the System to be heuristic in that as the System is used, the messages contained in the centralized diagnostic and tutorial files can be maintained and enhanced by the System maintainers.

4.1.6 Automatic documentation. With the System concept of dynamic generation of simulation models, it is important that the generated models be fully documented so that it is clear how to use them. To attain this goal, the model building portion of the Executive, upon demand by the user, will automatically generate documentation concerning the model. This documentation will have two purposes: to describe the construction of a model, and to describe how to use the model. That is, the documentation will be detailed enough to explain the algorithmic base of the model to the engineer, and will be clear and concise enough about the use of the model to make it amenable for use by the engineer with the problem under study. To understand how the System can automatically produce documentation about an SSM, it is important to understand the data that are stored within the TMs in the library of the model building System. Table 4 describes the preambles of each of the types of data stored within the System libraries. It is from these data that the System generates the automatic documentation.

Table 4. System Library Preambles

SOFTWARE MODULE	FLOW SCENARIO	SSM
NAME	NAME	NAME
SIZE	SM ID	SM ID
CALL PARM	STM ID	STM ID
CRID	SS ID FLOW ORDER	SCENARIO
INPUTS	TEXT DESCRIPTION	FILES
OUTPUTS		CORE
TEXT DESCRIPTION		CRs

STM	FILES	TECHNOLOGY MODULE
NAME	NAME	NAME
INPUTS	ITEMS	TEXT DESCRIPTION
OUTPUTS	ID	LOGIC TABLE
TEXT DESCRIPTION	SIZE	
CALL TABLE	FORM	
	FILE SIZE	

COMMON ROUTINES	INTERFACE TABLE
NAME	TM NAME
SIZE	INPUT FORM
CALL PARM	OUTPUT FORM
INPUT	
OUTPUT	

4.1.6.1 Mathematical basis documentation. The primary input source to the model building Executive for the development of automatic documentation that defines the mathematical basis of the SSM being built is contained in the software module preambles, the flow scenario preambles, and the STM preambles.

Contained within the software module preamble is a textual description of the function of that module, which was prepared by the module developer. This information, as well as any common routines from the math pack used by the module, is output by the model building Executive. The preamble to the STM also contains a textual description of the STM which is supplied by the model building user when an STM is described. Additionally, the System generates a call table that is output to the user as part of the automatic documentation, which describes the exact flow of the STM. Since the STM is a subset of an overall TM, a flow diagram depicting the software modules and the flow among them is generated by the model building Executive from the call table and included in the user documentation.

The flow scenario preamble contains a description of all technologies present in the SSM and all the software modules comprising the SSM. The flow table generated by the System describes the exact flow of the entire SSM and the conditions upon which different processing paths may be taken during the execution of the SSM.

It is from these three sets of data that the mathematical basis of the SSM can be described. If the engineering user wishes more detailed information, then he may peruse the detailed documentation that is available from the SSM builder function. This includes a complete listing, in structured FORTRAN, of the SSM requested. This is possible since an SSM can be constructed from the source code of TMs and compiled during the model building phase of the System. This listing would be comparable to the listing developed by a programmer if the SSM were defined as a project and were not built by a model building System. Therefore, to determine the mathematical basis of a particular SSM, the user has at his disposal as much documentation as would be available for any program developed as a separate project.

4.1.6.2 Automated user documentation. The user-oriented documentation that is automatically output

by the model building System during the development of an SSM is based upon the user's manual as defined in DoD Manual 4120.17M. Table 5 is an excerpt from DoD

Table 5. Sample User's Manual Table of Contents

TABLE OF CONTENTS

Manual 4120.17M, and defines the table of contents for a user's manual. The model building System will output, during the final phase of an SSM build function, a user's manual in this format, which will allow an engineer to use the model generated without referring to any other System documentation.

Section 1 of the manual will be predefined and stored in the System library and will be the same for all SSMs. Section 2, System Summary, will contain all seven required paragraphs describing the SSM that was built. System Application will be defined by using the developer's textual descriptions of the software modules and the STM builder's textual descriptions of each STM. These descriptions, in total, describe the purpose of this model and its application.

The System Operation section will describe how the engineer initiates the SSM by invoking the JCL that was generated by the model building System and stored as an integral part of the SSM. The System Configuration section will contain the hardware description and support software required to execute this SSM. The System Organization diagrams generated by the model building Executive will be built from the Scenario table, which describes the flow among the TMs. This will describe to the user the sequence of events that will happen during model execution as the simulation flows from one STM to another, based upon the solution technique that was described by the Model Builder.

Performance data will be defined by knowing the size and the flow of each STM and estimates by the System will be output in this section of the document describing the general requirements for performance of the model. The Data Base section will contain the names of all files that are external to the model and are required for its execution. The final subsection in System Summary will contain a description of each input required of the user and a list of all outputs that the user can expect to have generated during model execution. Therefore, Section 2 of the system-generated SSM user's manual will be as complete and concise as the user's manual written by users specifically for this SSM.

Section 3, Staff Functions Related to Technical Operations, basically tells the user how to input the data he needs and how to get his output from this particular model. All nine required subsections of Section 3 will be developed by the model building Executive and will be included in the SSM user's manual. These sections are self-explanatory. It should be noted that the System will put in the SSM user's manual, in accordance with Sections 3.5 and 3.8, samples of each of the inputs and outputs that the user is requested to give and can expect from the System, respectively. Section 4, File Query Procedures, will be dedicated to the interactive capabilities that were built into a particular SSM. It will describe pauses that have been built into the model during SSM definition that will allow the SSM user to review intermediate results of model execution and to input new data required for continuation of the SSM run.

4.1.6.3 Automatic documentation summary. This section on automatic documentation has described, in generic terms, the documentation that will be produced automatically by the model building Executive during the SSM build function. This documentation has two purposes. The engineering user can examine the documentation for the model he wishes to use to find: its mathematical basis, the algorithms contained in the model, the flow of the model, and the level of complexity he can expect to be applied to his engineering problem. Secondly, a complete user's manual, based on DoD standards for project development, is produced by the model building Executive for each SSM, containing all required subparagraphs as previously described.

4.2 How the engineer will use the System.

4.2.1 Preparing for use of the System. The engineer will prepare for use of the System by reviewing documentation of the System as it exists on his host computer system. The engineer will have a particular problem to be solved involving "aircraft" technical, physical, and operational characteristics at a given level of complexity. The System has three major phases of use:

- Model Building
- Model Execution
- Input and Output Processing.

In the simplest cases the engineer will be involved in only the second and third phases. He will review the collection of SSMs that exist in the library at his installation, and see if the capability of one of these matches the definition of his problem. If one these SSMs does match his problem, he then reviews the documentation to determine input requirements and output options for that SSM. The engineer will specify the input required by:

- Identifying input data files from the library
- Providing input by cards or tapes
- A combination of the above

New input data files may be created or old data files may be edited either in a batch or interactive mode. The user will then submit his job for execution.

4.2.1.1 Input data preprocessing. The engineer may elect to take one additional preparatory step prior to initiating execution of a model. This step deals with the preliminary validation of model inputs. In instances where System standard input data have been modified or where totally new input data have been developed, the engineer will utilize the input data validation functions of the Executive data handler to print, plot, or perform data range checks for selected input groups.

4.2.2 Model use (SSM execution). The SSM may be executed in two ways via two different modes. The user may elect to execute only to the point where control inputs are reviewed for completeness and displayed in graphical and tabular form to determine accuracy of input data, or the SSM may be executed to completion. The SSM may be run in a batch mode or in an interactive mode. If the host system has interactive capability, input data may be reviewed for completeness and accuracy, and execution of the job may proceed after any required editing of input data. All computed results that are normally saved for output processing will be available after execution unless control input has been provided to override defaults for saving

computed results. In addition, if control inputs have been provided to save more detailed results than normally saved, these results will also be saved for normal output. Input for output processing will then control the data actually output and the form of that output; output data may be:

- Tabulated
- Plotted
- Assigned to a permanent storage file for later use

With an interactive graphics terminal, results could be displayed in tabular and/or graphical form for review. Decisions could then be made about modifying data files and execution of another job.

4.2.3 Model building.

4.2.3.1 New SSMs from old SSs. If, after a review of documentation of the SSMs existing in the library, the engineer finds that he does not have an SSM that matches the requirements of his problem, he may take a second look at what is available in the library. The library will contain Specific Scenarios that have been used to develop SSMs. The SSs are calling sequences for the execution of STMs. A new Scenario may be developed by editing an existing SS (or a completely new SS may be developed). The new Scenario is developed using the CHARLES language and is based on existing STMs. The engineer may then define input to create a new SSM from the new Scenario. He may give instructions (using the CHARLES language) to have the Scenario saved; it would then become an SS and this set of CHARLES instructions would be saved for future use. The SS and the corresponding SSM that are created are given a unique identifying name for future reference. The Model Builder provides descriptive information (the user-defined narrative) to be stored with the SS and SSM. Automatic documentation is produced that defines the STMs making up the SSM, input requirements and options, output options, estimates of execution time, and resource requirements. This new SSM may then be executed as outlined above for other existing SSMs.

4.2.3.2 New STMs. If TMs are available that provide the technology desired by the engineer, but not the desired STM which gives him the capability that he desires, the engineer may create new STMs , edit an existing SS and incorporate the new STM, or create a completely new Scenario that includes the new STM. The System may then be used to create a new SSM from the Scenario and the engineer may choose to save the Scenario as an SS as outlined above.

4.2.3.3 Adding new technology. New technology may be added to the System by two methods:

- Adding new SMs to existing TMs
- Adding completely new TMs

In both steps, the engineer will have to develop the new technology, have software modules coded into subroutines, and have these entered into the System library.

4.2.4 Input and output processing. The set of all possible outputs for an SSM are defined implicitly during the model building phase. For any given TM, there is a predefined set of potential output groups. Each output group is associated with one STM process (software module). When an STM is created from a TM, any output groups associated with processing options that are excluded are set to null (disarmed) in the STM output list. The next level of output controls is set in the Scenario creation process. When the engineer creates the controlling SSM Scenario or a Specific Scenario, he may disable (or enable) any armed output group by clearing (or setting) System control items that are defined for each non-null (armed) output group. By utilizing this secondary method of enabling and disabling output groups, any subset of an STM's possible outputs may be selected at model execution time by conditioning the output disables and/or enables upon externally supplied control variables. This output selection approach will be used in implementing the System PFCs, which will be supplied with the initial System releases.

The set of inputs required for an STM is also defined in terms of groups. For a given TM, specific input groups are associated with individual TM processes (software modules). As in the case of an output group, the creation of an STM from a TM determines which input groups are to be armed and which are to be nullified (disarmed). For example, one of the Airloads TMs could provide alternate processes for determining aerodynamic coefficients. If, in creating a STM for Airloads, the engineer elected to exclude the option for a table lookup of CL, CD, and CM in favor of calculating CL, CD, and CM from linear coefficients, the input group(s) associated with the table lookup process would be disarmed (i.e., the potential input group(s) for the table lookup process are nullified). All other input groups required by the STM are set to the armed and enabled state, where "enabled" in this context means that a particular input group requires external input data. A second level of input requirements reconciliation takes place during the Scenario creation phase of model building. For each STM referenced in a Scenario, the armed/enabled input groups are reconciled with predecessor STM armed output groups, and where matches are found the input group is set to the disabled state (meaning that no external input is required for the group). If the Scenario is defined as a Specific Scenario, all armed/enabled input groups are listed for the model builder and are included as part of the SS definition. Therefore, if a Scenario contains a reference to an SS, the input/output group reconciliation is handled in the same manner as that described for STMs, i.e., the SS is considered to have a single set of input and output groups.

4.2.4.1 Default model outputs. The default output from any SSM consists of all TM-defined output groups that are not implicitly excluded in the STM creation process minus all output groups which are explicitly excluded (suppressed) in the Scenario or Specific Scenario creation process of model building. In terms of the output processing described in Section 4.2.4, the default output for an STM consists of all output groups that are armed and enabled. Therefore, when the engineer executes an SSM with no externally supplied output controls, all armed/enabled output groups are written, in a System-defined format, to intermediate System output files. These output data are then available for post-processing by a subsequent job step or an independent data handling job.

4.2.4.2 Selecting specific outputs. For every output group defined for the CHAS, there will be a default output format. For commonly utilized output groups, optional output formats (tabular and/or plotted) will also be defined as System standards. When the engineer executes an SSM for the first time or when new technical, physical, or operational input data are being used, the engineer will select one or more specific output groups to provide insight into the overall quality of the simulation run and also to provide the basis for the selection of additional output groups, different output formats, or areas where only a specific time series of data is wanted for further postprocessing. This initial "quick look" data handling process would not be necessary for very simple models when the number of possible outputs is small or where most outputs were suppressed. The same would be true of more complex models where multiple runs of the same model are being made in order to refine data for the item under analysis.

4.2.4.3 Defining tailored outputs. In instances where System-supplied output formats for an output group or groups do not satisfy a particular reporting requirement, the engineer may define a tabular list format or (for arrays) a plotted output that is tailored to his specific requirements. When a new output format is defined, by using the output template creation function of the System data handler, the new format is saved as a user-defined output template in the CHAS library and is linked to the specific output group(s) that it accesses. From the time the new output template is defined until such time that is purged from the library, the user-defined template is listed as a user-supplied output option for the specified output group(s).

5. HOW THE SYSTEM WILL BE DEVELOPED

5.1 Organization responsibilities and structure. This section defines the responsibility of each organization involved in the Second Generation Comprehensive Helicopter Analysis System (CHAS) development. The structure, and interactions, are depicted in Figure 14. An explanation of the difference between the First Level and Second Level organization is presented in Section 5.1.2.

5.1.1 Organizational responsibilities. The organizations involved in the CHAS project are: (1) Applied Technology Laboratory (ATL), the Government Program Office, (2) Government-Industry Working Group (GIWG), (3) Technical Advisory Group (TAG), (4) Prime Development Contractor (PDC), (5) Technology Integration Contractor (TIC), (6) First Level Technology Module Developers, and (7) Second Level Technology Module Developers.

5.1.1.1 Applied Technology Laboratory. ATL has overall responsibility for the CHAS program and will have direct interface with the PDC, the GIWG, the TAG, and Second Level TM Developers. They will interact on an information exchange basis with the TIC and the First Level TM Developers. In addition, ATL will act with the prime development contractor to: review and approve all TM Type B5 development specifications; help in issuing RFPs; review results and proposals; determine TM developers with subcontract awards; and, approve TM acceptance criteria and final TM acceptance.

5.1.1.2 Government-Industry Working Group. The role of the GIWG is to enhance user orientation. It will meet a minimum of once per year to: review program progress; review general compliance with the Government and special industry needs; and identify errors, potential problems, and high-risk items within the CHAS. The unique function of this group is to allow helicopter manufacturers a voice in influencing the CHAS development, so as to gain acceptance and encourage use of the results and analysis and to provide guidance to each of the other participating CHAS development organizations.

5.1.1.3 Technical Advisory Group. The role of the TAG is to enhance the technical approach. It will meet a minimum of once per year to: review program progress; review compliance with the Government needs; and identify errors, potential problems and high-risk items within the CHAS. The function of this group is to coordinate and focus interested Government agencies toward the CHAS and to provide technical guidance to the other participating CHAS development organizations.

5.1.1.4 Prime Development Contractor. The PDC will have responsibility for all facets of the First Level Release including specifically: the development of the Executive, final integration of the TMs developed by PDC through PDC subcontractors, and integration and acceptance testing. The PDC is also responsible for defining integration requirements for Technology Module developers contracted directly by the Government, final integration of those TMs into the CHAS, and second level enhancements of the Executive. The PDC will provide for continuous quality assurance and ensure that all documentation is prepared in accordance with prescribed standards.

The PDC will have a full-time dedicated CHAS project manager responsible for five functions: (1) Executive development, (2) Technology Module coordination, (3) quality assurance, (4) documentation, and (5) contracts/ finance.

5.1.1.4.1 Executive development. Development of the Executive will be accomplished by seven development teams: (1) Model Building; (2) Language Processing; (3) Library Maintenance; (4) Subsystem Control; (5) Terminal Interface; (6) Output Processing; and, (7) Environment. Each team will be responsible for the development of one software segment of the Executive:

Team Number	Segment	Function
1	Model Builder	Development of the SSM
2	Language Processor	Interpret all user commands for all phases of the System
3	Library Maintenance	Maintain and enhance the entire CHAS

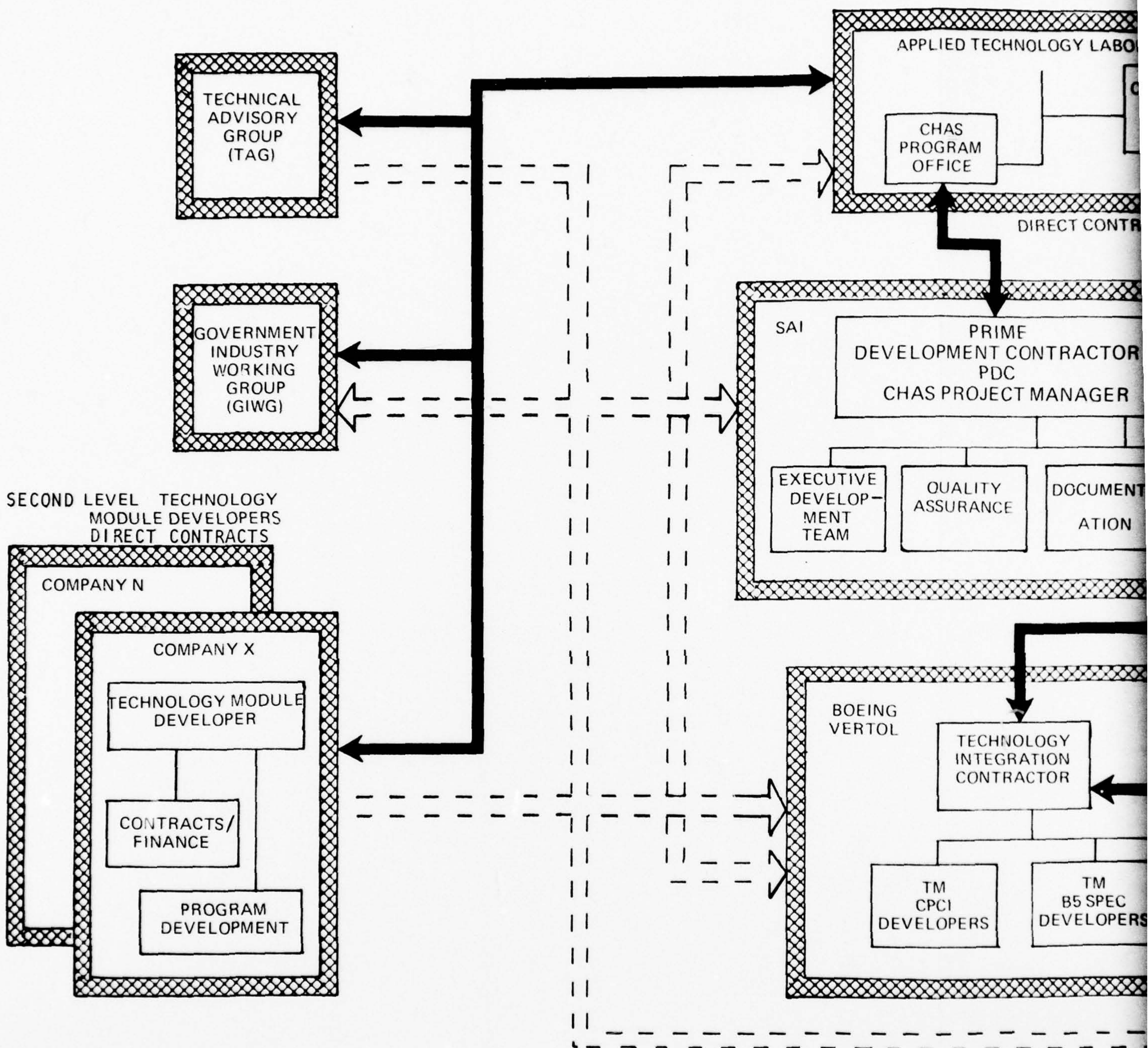
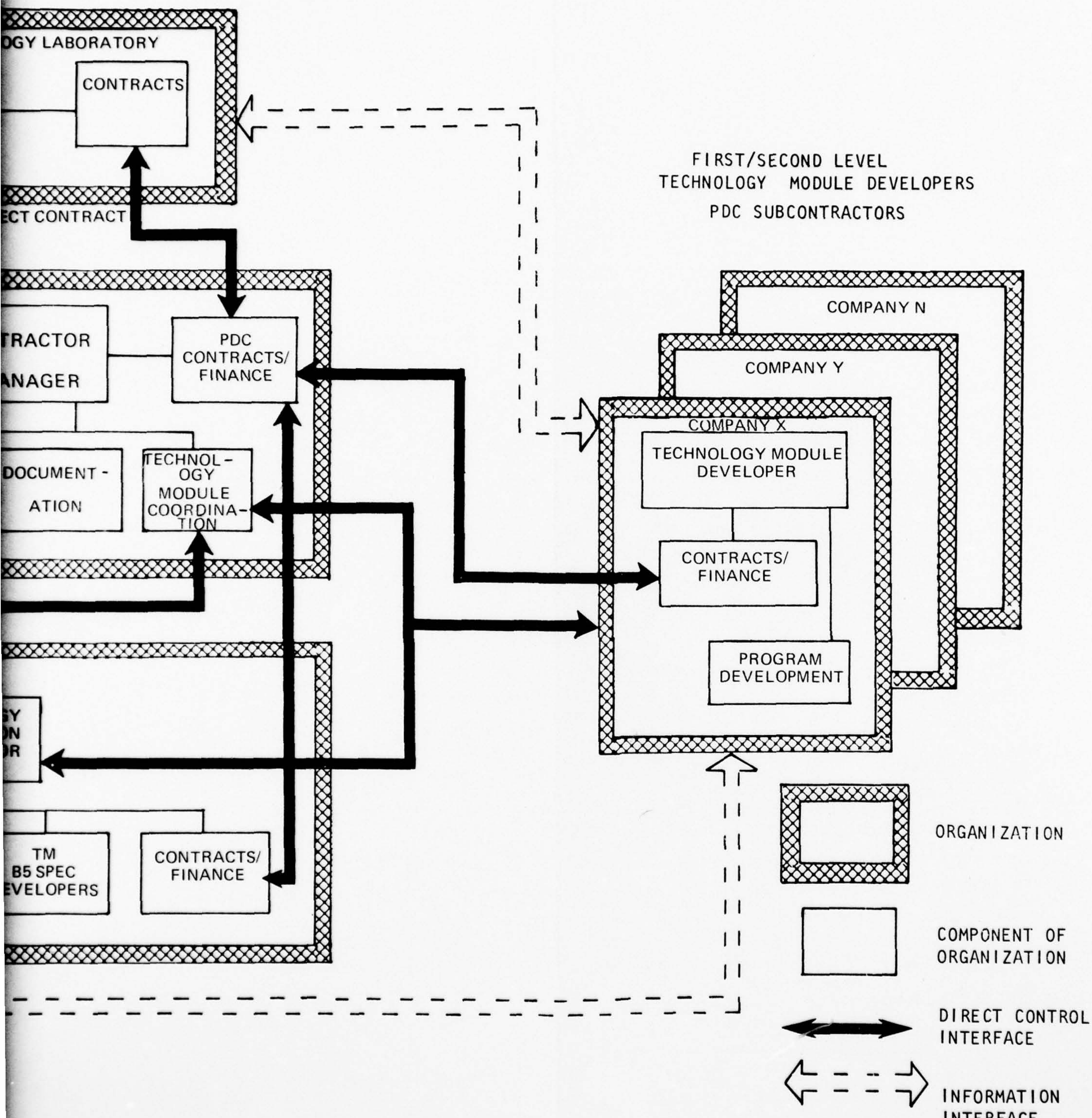


Figure 14. CHAS Organizational Structure



OGY LABORATORY
CONTRACTS
ECT CONTRACT
TRACTOR
ANAGER
PDC CONTRACTS/ FINANCE
DOCUMENT-ATION
TECHNOL-OGY MODULE COORDINA-TION
GY
ON
OR
TM B5 SPEC EVELOPERS
CONTRACTS/ FINANCE
FIRST/SECOND LEVEL
TECHNOLOGY MODULE DEVELOPERS
PDC SUBCONTRACTORS
COMPANY N
COMPANY Y
COMPANY X
TECHNOLOGY MODULE DEVELOPER
CONTRACTS/ FINANCE
PROGRAM DEVELOPMENT
ORGANIZATION
COMPONENT OF ORGANIZATION
DIRECT CONTROL INTERFACE
INFORMATION INTERFACE

2

Team Number	Segment	Function
4	Subsystem Control	Direct processing flow for all phases
5	Terminal Interface	Provide interactive processing for the CHAS
6	Output Processor	Provide presentation of all output from SSM Executive
7	Environment	Provide data base management and operating systems under which the CHAS will operate

5.1.1.4.2 Technology Module coordination. There is a strict division of responsibilities for the development of TMs. The teams devised for each TM consist of a requirement/integration team and an implementation team. The requirements/integration team will be a helicopter manufacturer (TIC) under contract to the PDC, and will be responsible for developing the baseline Type B5 development specification for a TM, including the module integration and acceptance requirements. Each TM will be developed as a separate project, and will be documented in accordance with DoD Manual 4120.17M. The implementation team for a TM may be the TIC, a PDC subcontractor, or a contractor working directly for the Government. For all First Level CPCIs, both teams will be responsible to the TM Coordinator, a full-time PDC representative. The PDC is fully responsible for all First Level TMs. The Government is responsible for ensuring that all provisions of the Type B-5 development specifications are met for a Second Level TM contracted for directly. Once the TM is accepted, the PDC is responsible for integrating it into the System.

5.1.1.4.3 Quality assurance. The PDC will have the responsibility for insuring quality assurance for the Executive and for First Level TMs. Quality assurance for each Second Level Technology Module will be the responsibility of the TM developer.

5.1.1.4.4 Documentation. The documentation products that will result from the CHAS development

phase are based on MIL-STD-490 and DoD manual 4120.17M. All documentation relating to the Executive will be the responsibility of the PDC. Documentation of a particular TM will be the responsibility of the designated TM developer. A complete description of the documents to be developed is contained in Section 5.3. The PDC will coordinate all documentation to ensure standards are consistently met.

5.1.1.4.5 Contracts/Finance. The PDC Project Manager will be responsible for negotiating the subcontracts with the TIC and the TM developers that are contracted by the PDC. All subcontracts will be subject to approval by ATL. He will also be responsible for monitoring the work of those organizations to ensure strict compliance with all provisions of the contract. In addition, he will administer all financial aspects of the contract and be responsible for making monthly reports of the work progress and financial status of the PDC and all subcontracts.

5.1.1.5 Technology Integration Contractor. The technology integration function will be performed by the helicopter manufacturer that is teamed with the Prime Development Contractor. As part of this function, the helicopter manufacturer will be responsible for:

- Defining units, sign conventions, names, coordinate systems, interfaces between components, etc., and enforcing uniformity and consistency in TM development, operation, and documentation

- Defining TM CPCI Type B5 development specifications, assisting in developing TM RFPs, assisting in evaluating proposals, assisting in awarding contracts, and interacting with TM developers to answer questions and assure compliance with specifications

- Interfacing with the Executive developer (PDC) and the TM developer, to keep current on progress, problems, etc., and to provide communication/ coordination between Executive and TM developers to ensure:

    - Consistency between the executive routines and TMs (this is needed to ensure that unexpected problems are solved consistent with the overall requirements)

AD-A064 289 SCIENCE APPLICATIONS INC MCLEAN VA F/G 1/3
PREDESIGN OF THE SECOND GENERATION COMPREHENSIVE HELICOPTER ANA--ETC(U)
DEC 78 T HAMRICK, D COPELAND, F TARZANIN DAAJ02-77-C-0059
UNCLASSIFIED USARTL-TR-78-42 NL

2 OF 2
AD
A064 289

END
DATE
FILMED
4--79
DDC

- Successful integration of the TMs with the Executive

- That warnings of potential or real problems are raised early, while they are still correctable

- Defining TM acceptance criteria, assisting the developer in understanding and meeting the criteria, and evaluating TM compliance.

To perform these tasks, the TIC will draw from experts within its company with the expertise necessary to write specific CPCI Type B5 development specifications, determine acceptance criteria, monitor progress, review documentation, assure compliance, answer questions, etc. The TIC program manager will coordinate these separate efforts and interface with the PDC, Government representatives, and TM developers.

5.1.1.6 Technology Module developers. TMs may be developed by the Government, by contractors working directly for the Government, by subcontractors to the PDC, or by the TIC, who is teamed with the PDC. The estimates contained in the plan for TM development can be used as guidelines for the evaluation of proposals for each TM CPCI. TM developers are responsible for designing, developing, and testing the TM consistent with the Type B5 development specification evolved by the TIC for the subject TM. The TM developer is responsible for all documentation, which will be in accordance with DoD manual 4120.17M.

5.1.2 Organization structure. Figure 14 depicts the organizational structure for the CHAS project.

5.1.2.1 First Level structure. For First Level products, ATL will have direct interface with the TAG and GIWG, and a single development contract with the PDC. The PDC will have a dedicated TM Coordinator, who will work with the TIC and the TM developers.

The Executive will be designed, developed, documented, tested, and demonstrated by the PDC. The TIC will develop Type B5 development specifications for all TMs and, through the PDC, will submit the specifications for approval by ATL. After approval, the PDC and TIC will recommend whether the module should be developed by the

TIC or contracted to another TM developer. Upon a decision by ATL, the PDC and TIC will issue an RFP for the development of the TM. The successful bidder, who will be determined by a joint decision of ATL, PDC, and TIC, will be responsible to the PDC under the technical guidance of the TIC. The PDC will accept the module from the TM developer and present the module for acceptance to ATL. The PDC will then integrate the TM into the System.

A direct information flow will be established among the TAG, GIWG, PDC, TIC, TM developer, and ATL.

5.1.2.2 Second level structure. The Second Level structure is identical to the first level structure with the exception that the Government may develop TMs or contract directly for TM development. In either case, the TM will be based upon Type B5 development specifications, developed by the TIC.

This addition of direct TM development contracts does not preclude the TIC or PDC subcontractors from continuing TM development through the Second Level System. For TMs contracted directly by the Government, acceptance will be the responsibility of ATL. Integration of the accepted TMs into the System remains the responsibility of the PDC.

5.2 Development activities. This paragraph describes the activities that are necessary to design and implement the CHAS. The description of these activities is dependent on a hierarchical organization of the components of the CHAS, which is depicted in Figure 15. At the highest level of this hierarchy is of course the System. The System is divided into parts called subsystems. The subsystems of the CHAS are the Executive and the various TMs. Some subsystems are then further subdivided into parts called software segments. The Executive has had software segments defined by the predesign effort; the TMs have not but may when they are developed. Software segments and subsystems that do not have software segments defined are made up of programs. A program is the lowest level for which documentation is defined. The structure depicted in Figure 15 is meant to be representative only. It is used for organizational purposes and does not imply functional flow.

In the following paragraphs the proposed methodology for developing the System is described in detail. The

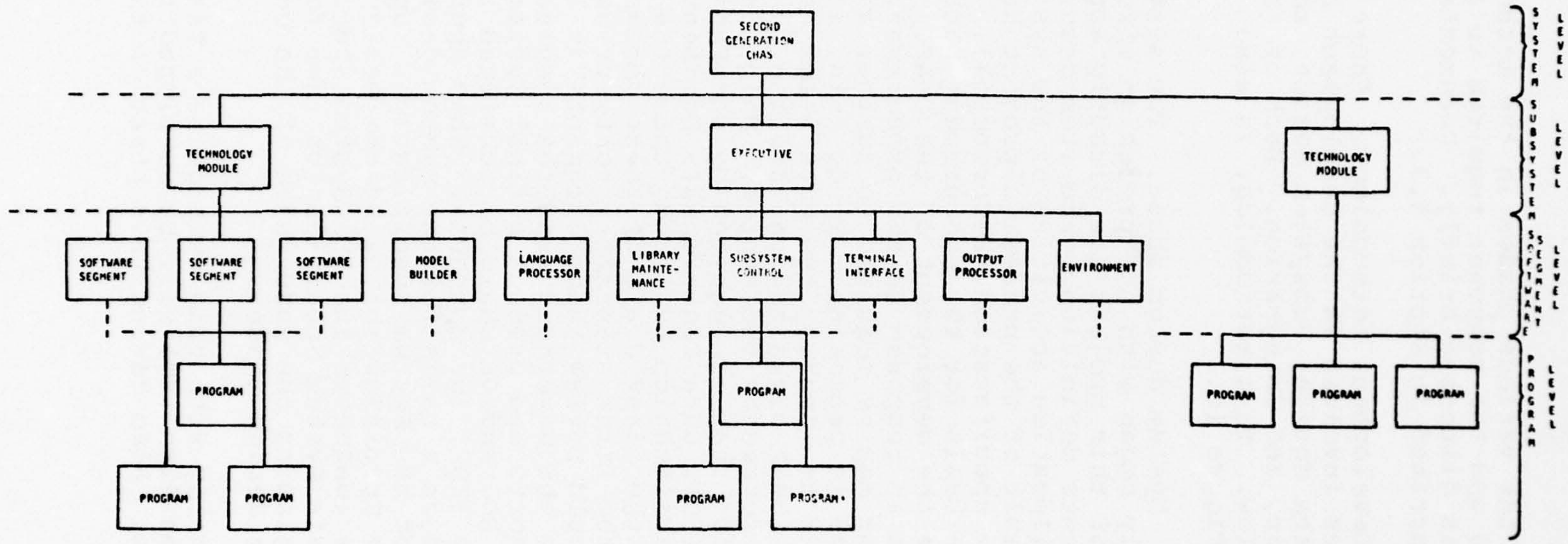


Figure 15. Hierarchical Organization of the CHAS

schedule for the various phases in the methodology is also provided and the manpower required to accomplish the project is discussed briefly. Documents mentioned are fully described in Section 5.3.

5.2.1 Development methodology. There are four primary phases involved in the development of the system: system design, subsystem design, software segment design, and integration. Each of these is described below. This methodology is also depicted in Figure 16.

5.2.1.1 System design phase. The system design phase actually began with the predesign effort, since the results of this project are directly applicable to the requirements definition, data item description, and functional allocation activities of the system design. The major result of the predesign project has been the Type A system specification (Reference 3). This document will form the basis for the procurement package that will initiate the development of the CHAS. In preparing the proposals in response to the procurement package, the contractor may be required to further define the system that he is proposing. To do this, it will be necessary that he identify the data items necessary for this system, their organization, and the functions that are to be performed by the system. This, in fact, is the information that is required by the Functional Description (FD), Data Requirements Document (RD), and Data Base Specification (DS) and, thus, the proposal will contain the first draft of these documents for the System. As the first step after contract award, the contractor would review these documents in light of the comments from the Government and the results of the predesign efforts and produce a final System FD, RD, and DS. The FD, RD, and DS would be submitted to a Functional Design Review (FDR) for approval. When approved the FD would be used as a basis for the System design and the development of the System Specification. Upon completion of the System Specification, a System Design Review (SDR) will be conducted and the System Specification approved. This System Specification then forms the basis for beginning the next phase of the development: the subsystem design phase.

Concurrently with this next phase a Test and Implementation Plan (PT) will be developed for the entire system. Also during this phase of the project

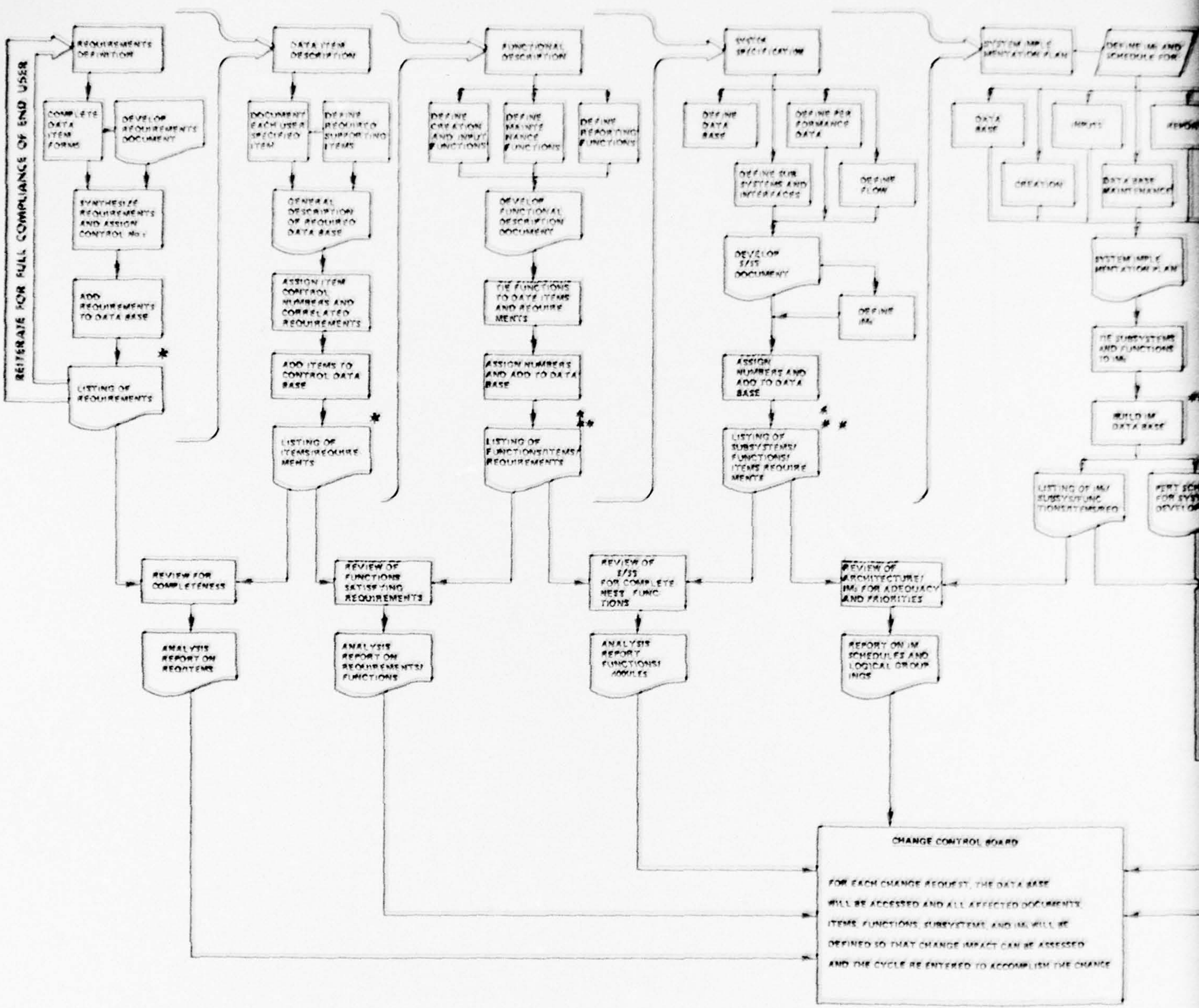
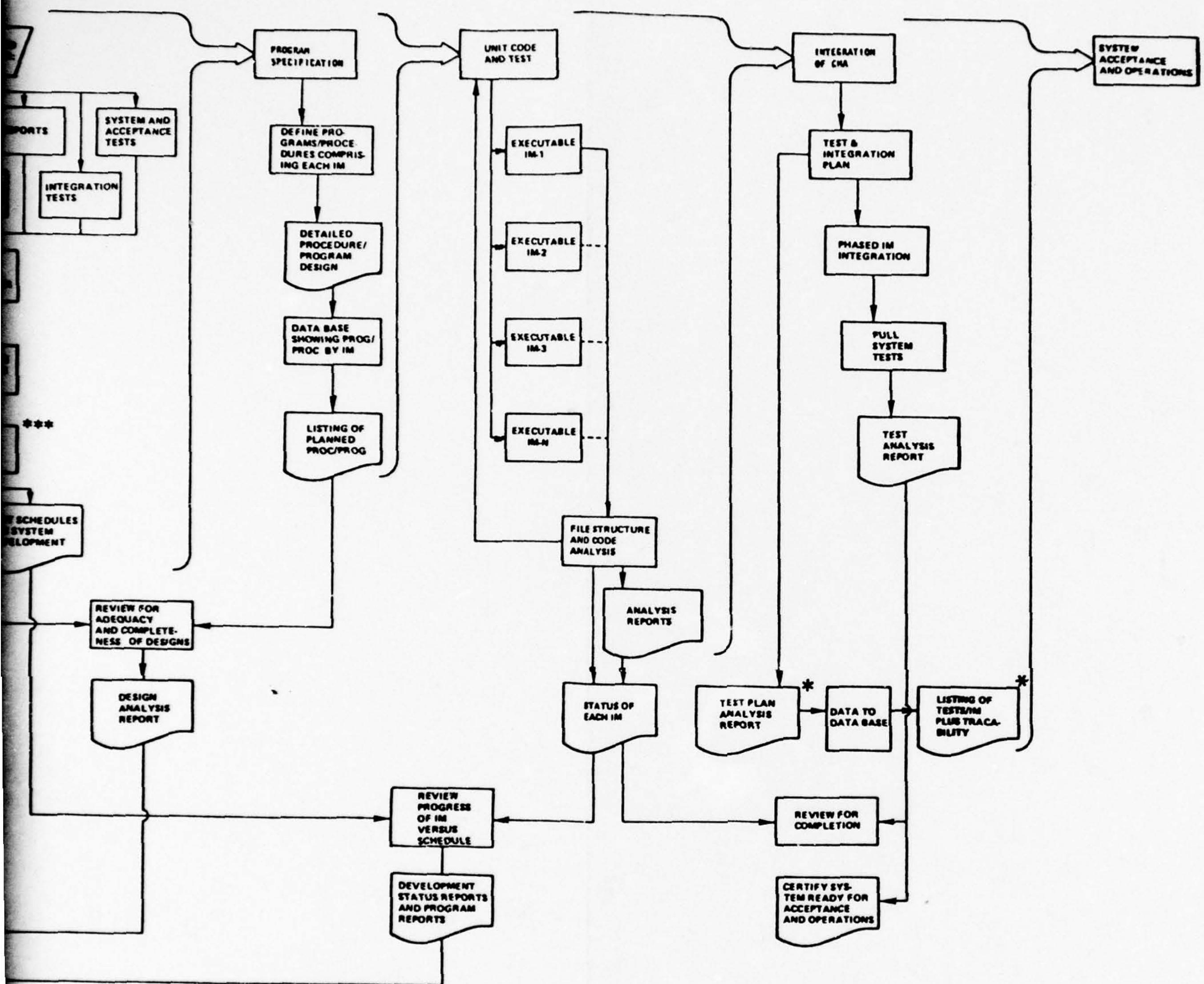


Figure 16. CHA Detailed Software Segment Development Methodology



* NEXUS - AN SAI MANAGEMENT INFORMATION SYSTEM

** SIZE AND COST ESTIMATION TOOL

*** STATUS DETERMINATION PROGRAM

2

the User's Manual and Computer Operations Manual will be started. Work will be continued on these two manuals as the System is developed so that at the end of the development activity they will completely represent the System. Both a UM and OM must be released with the First Level System. These documents will be continually updated as further releases are made and will be finalized by the Second Level System Release.

5.2.1.2 Subsystem design phase. After approval of the CHAS System Specification, the requirements allocated to the various subsystems (the Executive and the various Technology Modules) must be broken down in greater detail. In the case of those Technology Modules that are going to be separately procured, this process would result in the production of a Type B5 development specification for each TM in accordance with MIL-STD-490. A Type B5 document will form the basis for the procurement of each TM. During the development of the proposals as a result of this procurement, the bidder would perform much of the data item identification and function identification necessary for the FD, RD, and DS. Therefore these documents can be required to be produced at an early stage in the development contract of the TM. The PDC would, of course, perform these activities for the Executive. In any case the FD, RD, and DS would be submitted to an FDR for approval. Once approval is received, the System Specification for the subsystem would be developed. This System Specification would define the software segments that would make up the subsystem. In the case of TMs that have no software segments, the activities for developing those subsystems would be similar to those of a software segment as described in Section 5.2.1.3.

After completion of the System Specification, the document would be submitted for approval to a System Design Review. Upon approval, the next phase, software segment design, would begin. During this phase, the subsystem Test and Implementation Plan would be developed. As the software segment design and implementation phase progresses, the subsystem documentation will be continuously updated to reflect any changes that may develop due to the further definition of the System components.

5.2.1.3 Software segment design and implementation phase. After approval of the System Specification for the subsystem, the requirements allocated to each software segment can be broken down into even greater

detail. All data elements to be used by the software segment would be defined completely and their organization determined. All functions necessary to satisy the requirements as defined would also be identified. These activities will result in the FD, RD, and DS for the software segment being developed. These documents would be submitted for approval to a Functional Design Review.

Upon approval of the FD, RD, and DS, the work can begin on finalizing the Subsystem Specification. Upon completion of the Subsystem Specification for a software segment, it will be submitted to a System Design Review for approval. The software segment subsystem specification will, of course, identify each program that makes up the software segment. After approval of the Subsystem Specification, work will be started on completely designing each of these programs and a Program Specification (PS) for each program will be written. Each PS, as completed, will be submitted to an appropriately constituted review panel to ensure that it completely satisfies the functions allocated to that program by the Subsystem Specification. After approval, the actual coding of the program can be started. The code would then be compiled and debugged. Although there is no formal testing at the program level, each program would be subjected to unit testing to ensure that the program functions as specified by the PS.

Concurrent with these activities, the Test and Implementation Plan for the software segment would also be developed. Upon completion of the coding, debugging, and unit testing of all programs that make up the software segment, this PT would be executed. After completion of the testing, a Test Analysis Report would be written. If necessary, changes and corrections would be made in the software segment code and the necessary tests repeated until the acceptance criteria for the software segment has been met.

Technology Modules that are not subdivided into software segments would be implemented in accordance with the steps outlined in this paragraph.

5.2.1.4 Integration phase. As the various software segments are completed and accepted, subsystem integration will begin. Once the integration has been completed, the subsystem will be subjected to the testing specified in the subsystem PT. After the testing, the subsystem Test Analysis Report will be

written. If necessary, the subsystem code will be corrected and the necessary testing on both the software segment level and the subsystem level will be repeated until the subsystem meets acceptance criteria.

As the various subsystems are accepted, System integration can begin. The procedures for System integration are similar to those for subsystem integration. After the System has been integrated, the PT for the System will be executed and an RT produced. Again, if necessary, the code will be corrected and testing at all three levels, software segment, subsystem, and System, will be repeated as required. This iterative process would be continued until the entire System meets the acceptance criteria.

There are actually two integration phases for the development of the CHAS. Complete integration and testing will occur for the First Level System Release. After the First Level System Release has been concluded, integration of new, or enhanced, software segments will be conducted as these segments are completed. In every instance, the steps are the same as described above. The result of these integrations would be interim system releases. A final testing phase would be conducted at the end of the development contract and would result in the Second Level System Release of the Second Generation Comprehensive Helicopter Analysis System.

5.2.2 Development schedule. Figure 17 depicts the overall schedule and milestones for the CHAS project. As shown in the figure, the First Level System would be completed at the end of the 24th month after the award of the contract. The first quarter of the year three would be used for demonstration of the First Level release. At this point, almost all of the functions of the Executive have been developed. Only enhancements to various software segments would be conducted in year three and year four of the development contract. However, only a rudimentary level of technology will be represented by the Technology Modules that are completed during year one and two of the contract. They would be supplemented during years three and four with further Technology Modules that will completely satisfy all requirements of the Second Level Type A system specification. During years three and four as each TM is completed, it would be integrated into the First Level System. At this point an interim release of the System containing the TM could be made. The entire Second Level

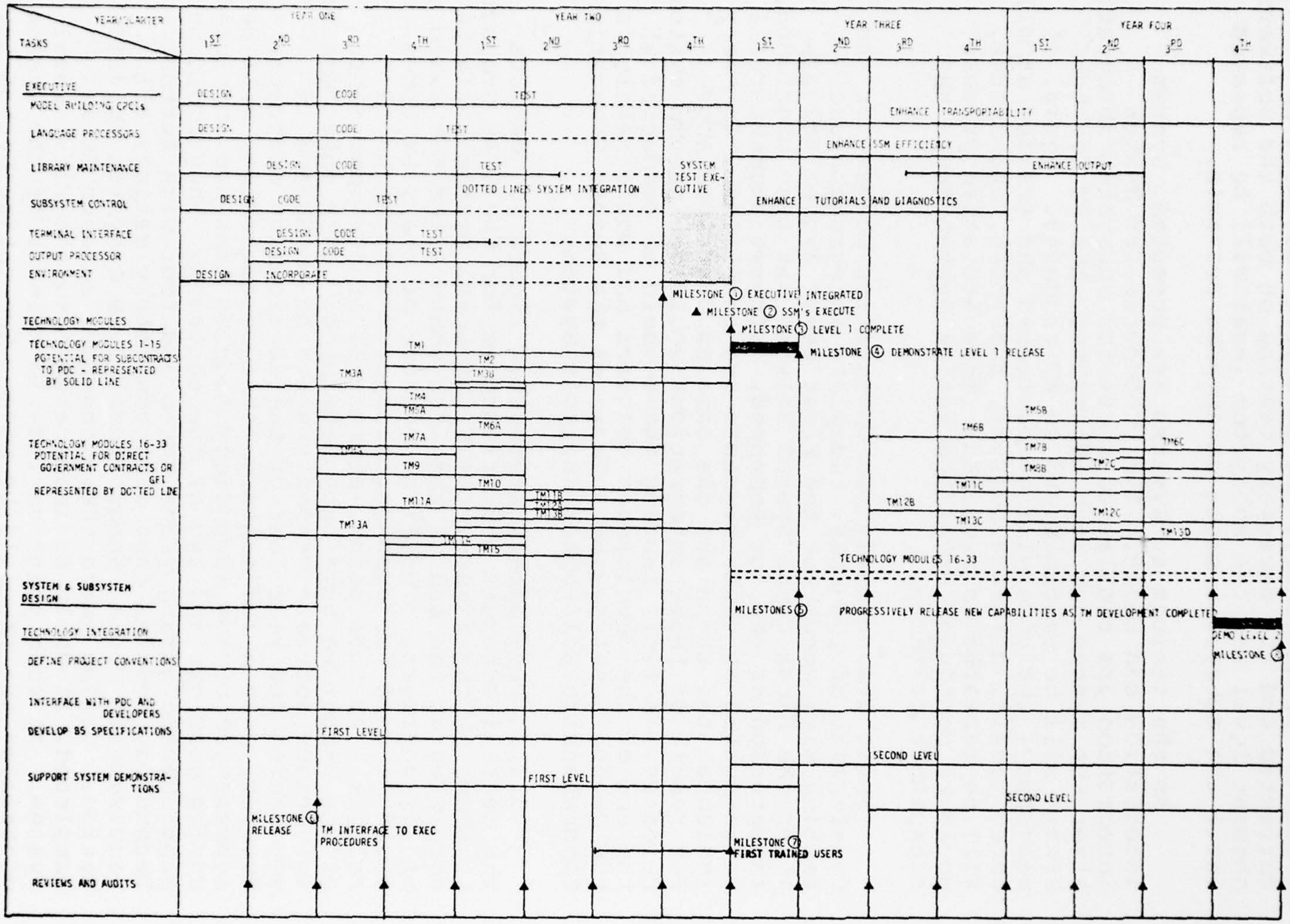


Figure 17. CHAS Overall Schedule and Milestones

CHAS will be completed by the 48th month of the development contract. The last quarter of the contract would also be used to demonstrate the Second Level CHAS and all of its capabilities.

5.2.3 Manpower requirements. The CHAS requires a fully operational Executive to generate the Specific Simulation Model before demonstrations can be accomplished. The resource allocation for the Executive development versus Technology Module development is as depicted in Figure 18. As shown in this figure, the effort expended on the Executive greatly exceeds that of the Technology Modules for the first 2 years of the development effort. However, this situation is reversed during the last 2 years. It may also be observed that toward the end of the first 2 years the manpower requirements are greatly reduced. The reason for this is that the TMs are to be separately procured and during the integration phase for the First Level Release no new procurements are anticipated.

The basis for the required level of effort for TMs for the First Level System is the amount of effort required to build a generic representation of each type of Technology Module in the System.

One of the primary advantages to this approach is that the System need not be released in just two major levels, First Level and Second Level. Instead, after the basic Executive is delivered at the First Level Release, the System can be enhanced with the development and integration of each Technology Module; thus, more capability is obtained sooner. The schedule for these releases is dependent upon ATL's selection of TM development.

The schedule and resource allocations satisfy virtually all Long Range System requirements that are within the current state of the art. To obtain this capability, it is estimated that approximately 1087 man-months (90.6 man-years) will be required. This is approximately 13 percent more than the project premise of 80 man-years. The allocation of resources among the required CHAS functions are in consonance with the SAI/BV design philosophy. Figure 18 depicts these allocations. These figures and other manpower estimates present only the technical engineering effort required and do not include documentation or administration support.

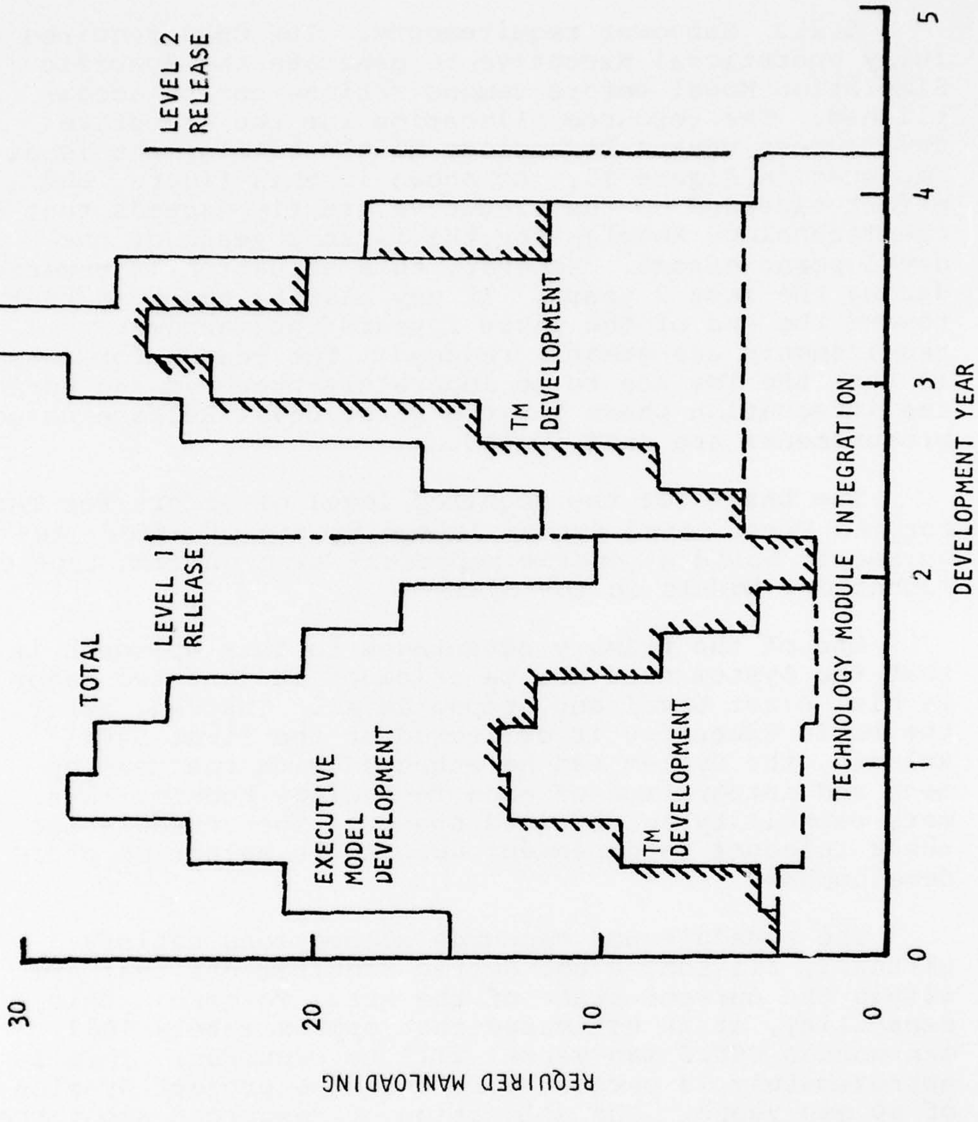


Figure 18. CHAS Development Effort

Figure 19 depicts each of the major development activities: First and Second Level Executive and Technology Module development, and technology integration. The estimates are in man-months by quarter and summaries for the project are in the totals column. There is also a summary by quarter along the bottom line of the figure.

5.3 Documentation requirements. Documentation of the Second Generation Comprehensive Helicopter Analysis System will be executed on three levels in accordance with the hierarchical organization of the system described in Section 5.2. The first level of documentation will be at the System level and will document the System as a complete entity. The second level of documentation will be at the subsystem level, that is, the Executive and Technology Modules, and will thoroughly document each of those subsystems. The third level of documentation will be at the software segment level. It should be noted that software segments have been defined for the Executive level but will only be defined for the Technology Modules as a result of the procurement action for those modules. All documents will be produced in accordance with either MIL-STD-490, or DoD Manual 4120.17M, as appropriate.

5.3.1 System level documentation. The Type A System Specification produced as a result of this predesign effort will form the basis for the procurement of the Second Generation Comprehensive Helicopter Analysis System (CHAS). The proposals received as a result of this procurement action, will essentially contain the information required for the Functional Description of the CHAS. The first document required under the development contract will be the Baseline Functional Description, which is scheduled for delivery at the end of the second month after award of the development contract. This FD would specify the complete final development schedule for the entire Second Generation CHAS and would detail the delivery dates of the other documents.

The other documents defined by DoD Manual 4120.17M which will be produced at the System level are as follows:

	YEAR 1				YEAR 2				YEAR 3				YEAR 4				TOTALS	
HELICOPTER MANUFACTURER - TECHNOLOGY INTEGRATION Define sign convention, units, etc., - Interface with software developer, write CPCI, acceptance, review, compliance, support full system integration and demonstration.	11	11	8	8	8	7	7	7	7	15	15	15	15	15	15	15	8	187
TECHNOLOGY MODULE DEVELOPMENT (FIRST LEVEL)		2	15	28	33	26	14	4										122
TECHNOLOGY MODULE DEVELOPMENT (SECOND LEVEL)											11	37	62	72	57	30	6	275
EXECUTIVE DEVELOPMENT (FIRST LEVEL)	36	52	52	52	44	44	42	41										363
EXECUTIVE DEVELOPMENT (SECOND LEVEL)									24	24	24	24	11	11	11	11		140
TOTALS	47	65	75	88	85	77	63	52	31	39	50	76	88	98	83	56	14	1087 man-mos 90.6 man-yrs

(Quarterly estimates in MAN-MONTHS)

First Level Release

Second Level Release

Figure 19. Overall Resource Allocation (MAN-MONTHS)

- System Specification (SS)
- Data Requirements Document (RD)
- Data Base Specification (DS)
- Users Manual (UM)
- Computer Operations Manual (OM)
- Test and Implementation Plan (PT)
- Test Analysis Report (RT)

The System Specification for the CHAS will specify the major components of the System and will describe their main functional capabilities. It will contain a general description of the Executive and each Technology Module, as well as the inputs and outputs for those modules. The RD will describe the data collection requirements for the CHAS as a whole and the DS will briefly describe the data bases that are involved or used by any component of the CHAS. The purpose of these documents is to provide a high level explanation or description, since corresponding documents will also be developed at both the subsystem and software segment level of documentation.

The User's Manual will contain all information required for the use of the CHAS. This manual would be divided into two distinct sections, the first of which would describe how the System is to be used by the Model Builder, and the second of which would describe how the System is to be used by the engineer. It must also contain sections describing the capabilities contained within each Technology Module, and how to build Specific Technology Modules from that Technology Module.

The Computer Operations Manual contains all information required by computer operators for execution of the CHAS during the model building and output phases. This information, as it relates to the execution of SSM, will be produced by the automatic documentation feature for each SSM.

The Test and Implementation Plan will contain the tests which are to be run prior to acceptance of the CHAS and the plan for the execution of these tests. At the conclusion of the testing period, a Test Analysis Report will be prepared. The purpose of the tests contained in this PT are to test the CHAS as a whole and should be written to ensure all requirements of the Type A system specification have been met.

5.3.2 Subsystem level documentation. There are two main categories of subsystems within the CHAS. The first is the Executive and the second consists of the Technology Modules. Documentation for each will be addressed separately.

5.3.2.1 Executive documentation. As a major subsystem of the CHAS, a Functional Description of the Executive is required. The complete list of documents defined by DoD Manual 4120.17M required for the Executive follows:

- Functional Description
- System Specification
- Data Requirements Document
- Data Base Specification
- Test and Implementation Plan
- Test Analysis Report

The SS should describe the Executive software segments. The RD and DS should present an overview of the data required for the Executive and its organization. The PT will contain tests that are designed to test the Executive against the requirements specified in the Executive FD. The RT will be the report of the execution of these tests.

5.3.2.2 Technology Module documentation. Each TM will be documented in the same manner as the Executive; that is, the following documents defined by DoD Manual 4120.17M will be required for each TM:

- Functional Description
- Data Requirements Document
- System Specification
- Data Base Specification
- Test and Implementation Plan
- Test Analysis Report

TMs may or may not be broken up into software segments. If they are broken up into software segments, then the above listed documents would have the same meaning as they have for the Executive. If the TM is not broken up into major software segments, then the SS must specify the TM completely and identify all programs that will make up the TM. For each program a Program Specification (PS) must

also be prepared. The RD and DS will also have to specify the data that is required and its organization and structure completely since there would be no lower level document. In either event, the PT must define tasks that adequately test the System and insure that all requirements specified by the Type B5 development specification, on which the procurement was based, have been met.

5.3.3 Software segment documentation. A software segment is a major portion of either the Executive or a TM. TMs may, or may not, have software segments. Software segments of TMs would be defined in the TM System Specification. Software segments of the Executive have been defined during this predesign contract.

Each software segment will be documented completely and for each software segment the following documents defined by DoD Manual 4120.17M will be developed:

- Functional Description
- Data Requirements Document
- Subsystem Specification
- Data Base Specification
- Program Specification
- Test and Implementation Plan
- Test and Analysis Report

The FD for a software segment will contain the requirements satisfied by the segment and the functions that are necessary for those requirements to be satisfied. It will also include a detailed development plan for the development and implementation of the software segment.

The Data Requirements Document will completely define all external data that is required by the software segment. This data may be external to CHAS or only external to this particular software segment. The Data Base Specification will describe completely the format and content of all data bases accessed by the software segment.

The Subsystem Specification of a software segment will completely define the logical flow of the software segment that implements the functions specified in the FD. It will define and briefly describe each program

that makes up the software segment. Each program identified in the Subsystem Specification will be further defined and detailed in a Program Specification.

A Test and Implementation Plan will be developed for each software segment. It will contain tests that completely exercise the software segment to determine that it satisfies all requirements specified in the Functional Description. It will also contain a detailed plan for the implementation of these tests. Subsequent to the execution of this test plan, a Test Analysis Report will be written to describe the results of the tests.

In developing the documentation for a software segment, consideration must be given to the fact that this contains program documentation, which is the lowest level of formal documentation.

5.3.4 Documentation change control. The CHAS is to be designed using a top-down approach. Therefore, the System-level documentation would be developed completely before developing subsystem level documentation. Documentation for a particular subsystem would be developed before any of its software segment documentation would be developed. However, as the lower level documentation is developed, that is, as the system is designed in more and more detail, changes will inevitably be made that will affect the high level documentation.

In order that these changes may be made in a controlled manner it is essential that each document that is developed and approved be subject to formal change control procedures.

As each document is completed, it will be submitted to a specifically formed board for approval. Upon approval, the document will form the baseline for the particular part of the System that it describes. Once the baseline has been established, any changes must be submitted to a change control board, and approved before inclusion. In approving documentation, it will

be the responsibility of the approval authority to insure that the document is consistent with previous system documentation which has already been baselined or that the appropriate changes have been submitted for approval. Further details of the precise procedures for implementing this change control are contained in Section 5.5.

5.4 Quality assurance requirements. An overall methodology within which the CHAS will be developed was defined in Section 5.2. This section defines how the application of this methodology, through a series of review processes, ensures satisfaction of project objectives. The reviews defined will be conducted at all three levels: system component, Executive, and CHAS.

Figure 20 depicts the review processes defined in this section. They will be conducted at the System, subsystem (Executive and TMs) and software segment levels. The review process as defined is possible because of the timeliness and accuracy of the data inputs that will be available from the PDC utilizing the project development tools defined in the Baseline Development Plan.

5.4.1 Functional Design Review. A Functional Design Review (FDR) will be scheduled at the completion of each Functional Description. The attendees for all reviews will be representatives from ATL, the PDC, TIC, and affected subcontractors. Representative input data for the FDR will be:

(1) Type A and B5 specifications
(2) Functional Description
(3) Data Requirements Document
(4) Data Base Specification
(5) A correlation report for Functions, Requirements
(6) Analysis Report

The analysis report will be based upon the NEXUS report and will define how each requirement is met and any requirements not satisfied. The results of this analysis will be presented by the PDC to the Functional Design Review Board. The preparation of all review input data is the responsibility of the PDC and minutes of the meeting will be developed by the PDC and will contain the ATL directed action items.

5.4.2 System Design Review. The purpose of the System Design Review (SDR) is to assure that the System, as designed, does satisfy the requirements. The input data will be:

(1) Functional Description
(2) System/Subsystem Specification
(3) A correlation report for Programs Architecture versus Functions
(4) Analysis report of correlations

Using input items (3) and (4) above, a structured walkthrough of the design will be presented by the PDC. Any problems will result in specific ATL directed action items. The results of the meeting will be documented by the PDC.

5.4.3 Change control review. Change must be considered as having a potential project impact during any phase of a development effort. Change proposals will be controlled in accordance with the CHAS Configuration Management Plan. Under this plan change control reviews will be held to determine resolution of all proposed changes. The input data available at each review will be developed by the PDC and will consist of:

(1) Correlation reports defining each requirement, function, and program affected, as applicable
(2) Analysis Reports

The analysis report would define the ramifications of the change concerning system capability, cost, and schedule. The PDC will present a recommended course of action for each change proposal. The change control authority will determine the course of action to be taken and authorize its execution.

5.4.4 Test analysis reviews. Test analysis reviews are conducted prior to testing for system, integration, and acceptance and are a critical review of stated test objectives, test conditions (environment, prerequisites, control parameters, personnel requirements), the functions to be tested, acceptance criteria, and the testing sequence. The primary purpose of the analysis is to determine if the tests are adequate for their specified purpose. Once it is

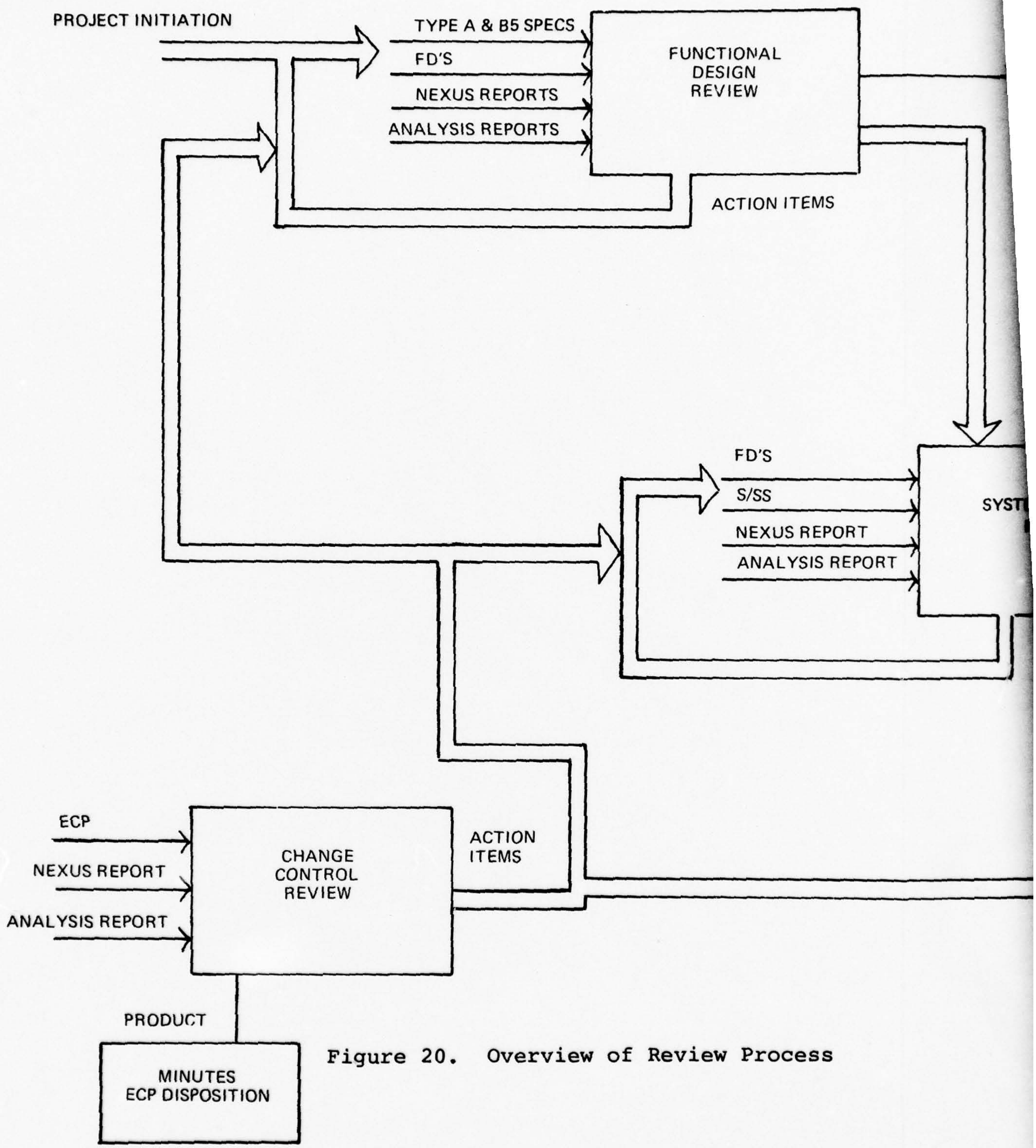
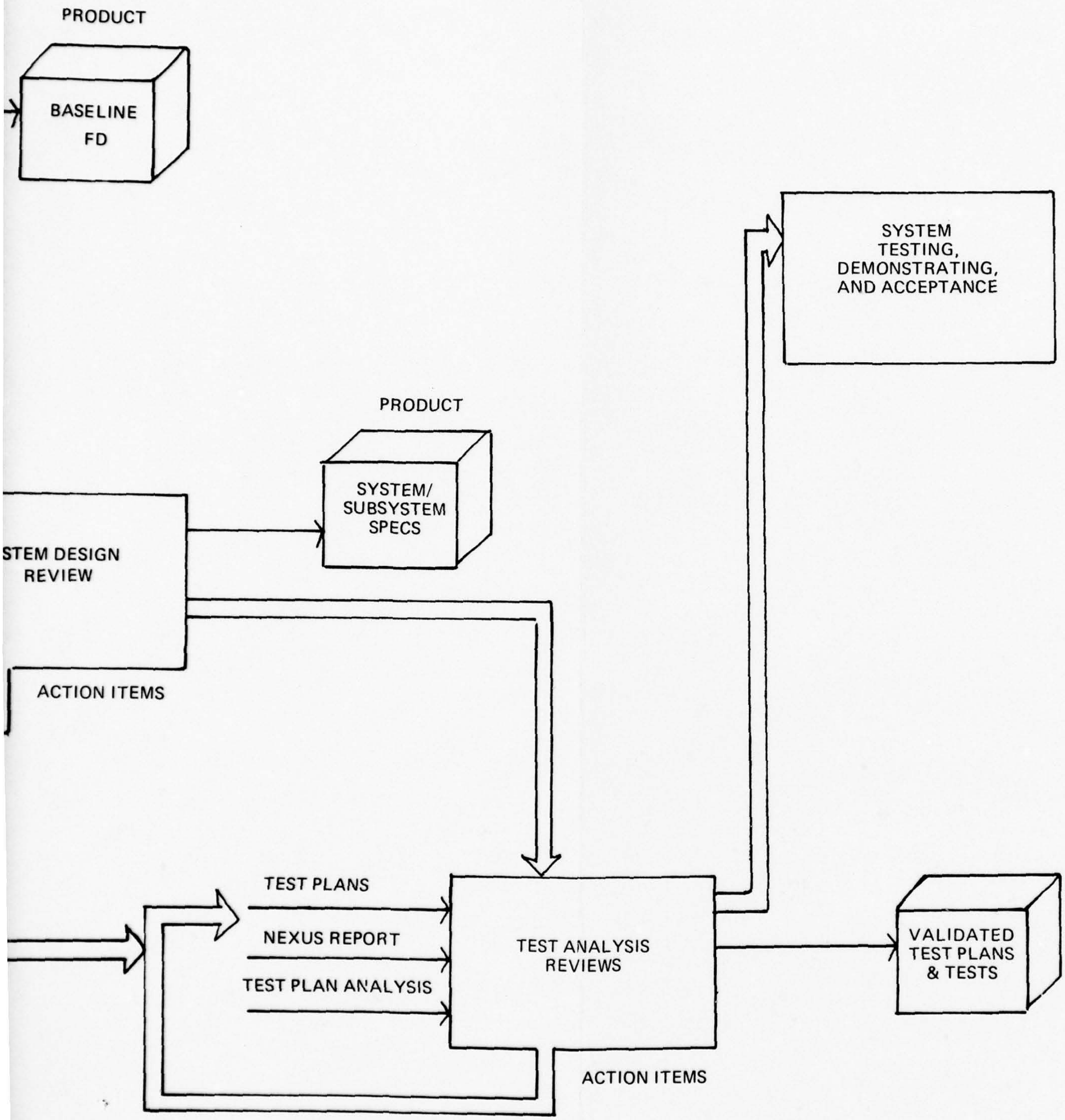


Figure 20. Overview of Review Process


PRODUCT
BASELINE FD
SYSTEM TESTING, DEMONSTRATING, AND ACCEPTANCE
PRODUCT
SYSTEM/ SUBSYSTEM SPECS
STEM DESIGN REVIEW
ACTION ITEMS
TEST PLANS
NEXUS REPORT
TEST PLAN ANALYSIS
TEST ANALYSIS REVIEWS
VALIDATED TEST PLANS & TESTS
ACTION ITEMS

concluded that the tests, when executed, will indeed demonstrate their planned function, a procedural analysis will address both the procedural and the functional adequacy of the test plan. Procedurally, the test plan must contain the steps required in preparation for the execution of the test, for initializing the test, any manual intervention that may be required during the test execution, and the instructions for handling or interpreting test results. It must also include details on what procedures are to be followed in the event of abnormal test termination, and it must give data reduction instruction.

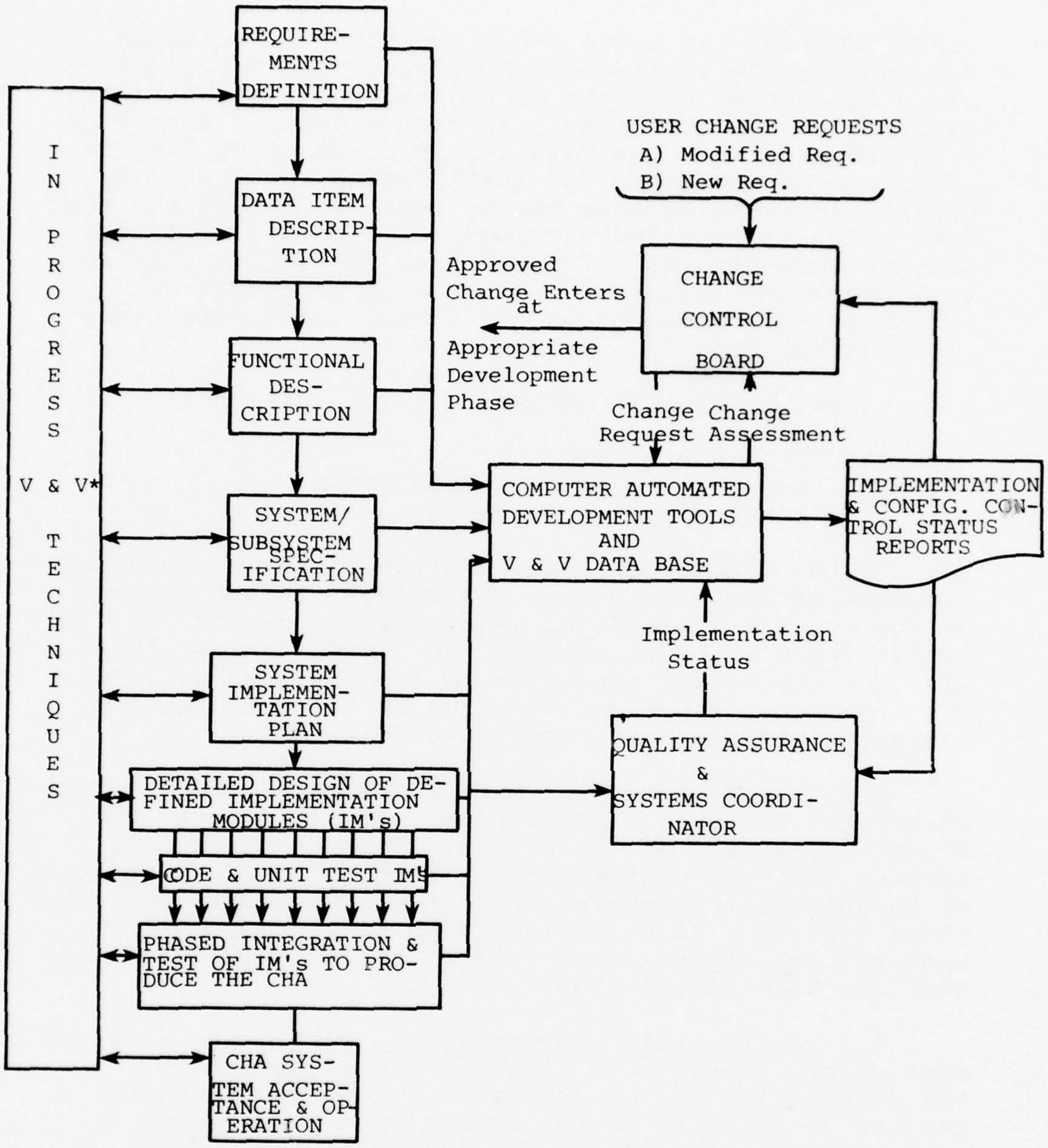
Input to the test analysis reviews will be:

(1) Test plans
(2) Correlation reports from NEXUS concerning test coverage

In the functional area, the test plan is evaluated in terms of the stated test objectives, in the context of the overall system test requirements, versus the functional tests to be performed. In simple terms: "What will the successful execution of the test, as defined by the test plan, represent in terms of the total system, subsystem, module, and program validation?"

5.4.5 Code reviews. During the code and unit test phase, the PDC will produce code analysis reports for selected portions of the system. These reports will include analysis of the coding technique used, adequacy of technique, and adherence to standards. Data for these reports will be derived from both programmer analysis and reports from various tools described in the Baseline Development Plan.

5.5 Configuration management. During the development phase, change authorization varies according to the disposition of the configuration item. Figure 21 depicts an overview of the control process. Documentation is baselined following its acceptance at the first scheduled review, i.e., an FD would be incorporated in the baseline when accepted at the Functional Design Review. The Change Control Board for the development phase will be comprised of ATL and its designated representatives, the PDC, and the TIC. All analysis for an engineering change proposal will be conducted by the PDC and presented at the configuration control review as



* Verification and Validation

Figure 21. The Overall View of the Control Process

defined in Section 5.4. Approved changes are incorporated and released to the standard distribution (defined by ATL).

Reports on all data concerning changes to program development configuration items will be produced and maintained by the PDC. Change authority to correct errors is at the sole discretion of the PDC after the CPCI has been presented for integration. At that time changes that affect the function of the program will be presented to the change control board for approval. Changes that correct a discrepancy will remain at the discretion of the PDC.

5.6. Testing requirements. Formal testing will be conducted on three levels: System, subsystem and software segments. The respective Test and Implementation Plans developed at each level will describe precisely the tests to be conducted and how they are to be conducted. The contents of these plans were described in Section 5.3. The following paragraphs define some special considerations for testing of the Executive, the TMs, and the CHAS as a whole.

5.6.1 Executive testing considerations. The ultimate criteria for determining acceptance of the Executive is its ability to produce a Specific Simulation Model corresponding to the user's problem definition, which will execute on the host machine, and its ability to produce the required model output. Additionally, the human factors that will make the System usable in the helicopter community must be acceptable. Therefore, for acceptance purposes, there are three types of Executive software segments: those involved in producing SSMs, those applying to human factors, and those required to support the Executive's execution.

The Executive software segments that produce SSMs will be defined as acceptable when the SSMs they build in response to a Particular Functional Capability (PFC) requirement successfully satisfy the PFC criteria.

The acceptance criteria for segments that provide System interface to the user require human factors consideration and will be defined by the PDC/TIC and presented to the ATL for approval. Upon approval, specific human factors acceptance tests will be developed and executed by the PDC.

Acceptance criteria for those segments that cause the Executive to operate will be defined by the PDC and their acceptance will be de facto upon successful acceptance of SSMs and user interface segments.

5.6.2 Test requirements for Technology Modules. A Test and Implementation Plan will be developed to describe types of tests to be conducted and test procedures to be used to guarantee that functional requirements for a TM have been met. The following types of tests will be conducted for each TM:

- Major function testing (TM developer's computer)
- CPCI testing (TM developer's computer)
- Acceptance testing (technology integration computer-final test)
- Preliminary validation testing

Both major functions and the complete TM will be tested for numerical and logical accuracy. All mathematical operations will be checked; end-to-end numerical checks of all major functions and the complete TM will be made. All coding will be checked against programming standards of Reference 8. Each major function and the complete TM will be tested with a range of data that forces the execution of all decision points and processing paths. Acceptance test cases will be established and run to demonstrate the processing and output capabilities of each TM. Preliminary validation of each TM will be conducted by performing the following types of evaluation of the results of each TM:

- Comparison of TM results with results from a similar analysis method
- Comparison of TM results with results from the exact solution of a classical problem (if applicable)
- Comparison of TM results with results from a physical test of a full scale or model helicopter or helicopter component

---

[8] PROGRAMMING STANDARDS FOR THE SECOND GENERATION COMPREHENSIVE HELICOPTER ANALYSIS SYSTEM, in preparation by Applied Technology Laboratory, U.S. Army Research and Technology Laboratories (AVRADCOM), Fort Eustis, Virginia.

5.6.3 System testing considerations. At the System level two problems must be considered: integration of new TMs with the current system and overall system acceptance.

The PT for a Technology Module shall include an integration test plan for the integration of the TM with the Executive that defines the set of tests necessary to validate: the data management (library maintenance) functions to catalog the TMs; the model building functions which provide for varying the software architecture of a TM and producing a STM; the model building functions that provide for combining specific TMs into executable SSMs; the support functions, e.g., checkpoint and restart, which are incorporated into SSMs; the data handling functions that perform the preprocessing and postprocessing functions of input data validation and output data reduction and formatting.

The CHAS PT shall be structured as a two-part plan. The first part shall consist of tests and demonstrations to verify that the CHAS satisfies the requirements for Particular Functional Capabilities as specified by the Type A system specification. The second part shall consist of operational evaluation tests designed to provide an objective measure of the CHAS viability in terms of: human factors, e.g., base of use, effectiveness in the engineering environment in terms of cost, setup time, and interpretation of results; adequacy of documentation (content, organization, and clarity) for both System use and System maintenance.

6. RISK ASSESSMENT OF THE PROPOSED APPROACH

6.1 Cost benefit. The SAI/BV proposed approach to a Comprehensive Helicopter Analysis System has great potential for being a useful program with growth capability over the next 15-year period.

The initial cost of developing the Executive has been discussed in Section 5. The payoff in flexibility (provided by the truly modular design) will occur over the entire life of the System:

- New technology may be incorporated easily at any time for future releases of the System.

- Individual companies or agencies using the System may add their own technology in the form of software modules.

- The helicopter analytical model may be reduced to the simplest valid model for the problem being considered to minimize computer execution costs and memory requirements.

- The System may be developed to be self documenting for Specific Simulation Modules (SSM's). This will provide program description and user input requirements/options and output options.

- Flexibility means that new helicopter configurations could be analyzed without developing entire new computer programs; turnaround time for developing new analysis capability would be shortened.

- Complexity of the System may be expanded in any direction including engine analysis, structural analysis, and aerodynamics; limits which might now exist on use of methods due to high CPU time (cost) or storage requirements are likely not to exist possibly 10 years from now if the historical trend in computer development continues. This System will permit taking advantage of projected computer development.

- In an interactive mode (Level Two Release), the System could prompt and teach the Model Builder and Model User; the Model User could review input data and intermediate calculations as well as end results.

The costs associated with the development of a Comprehensive Helicopter Analysis System can be put into perspective by considering the expenditures by the U.S. Army for development, production, operation, maintenance, and repair of helicopter hardware. The cost of development of helicopter analysis capability can also be measured against problems and associated costs that have occurred in areas of performance, vibration, and component failures in helicopter development programs sponsored by the U.S. Government in the past decade.

Data in Table 6 indicate that approximately 1.0 billion dollars have been requested by DoD departments for fiscal year 1979 for helicopter procurement, spares, and RDT&E. Cost estimates for development of the Second Generation CHAS are miniscule by comparison to the fiscal 1979 requests.

Table 6. Budget Requests for Helicopter Procurement, Spares, and RDT&E, Fiscal Year 1979 ($ million)

Dept.	Aircraft	Helicopter Procurement	Spares	RDT&E
Army	Sikorsky UH-60	346.3	30.6	3.0
Army	Bell AH-1S	136.9	3.8	10.6
Navy	Sikorsky CH-53E	168.1	15.1	-
Army	Hughes AH-64	-	-	177.4
Army	Boeing Vertol CH-47	22.6	-	19.5
Navy	Sikorsky Lamps	-	-	124.5
Totals		673.9	49.5	335.0

Total Procurement, Spares, RDT&E = $1,058.4 million

Development of the CHAS could have a significant impact on the quality, development cost, and operational costs of helicopters developed in the future and could lead to improvements in existing helicopters and helicopters already under development. In other words,

improvements in technical capability by development of the CHAS could yield a significant benefit relative to the development cost of the CHAS. The ground rules initially established a development cost of 80 man-years, spread over a 4-year period for the CHAS.

6.2 Critical items

6.2.1 Executive item. The SAI/BV system philosophy of building an Executive that produces specific models upon demand dictates that actual system demonstrations of technical solutions are possible only after the Executive can produce the model. Consequently, the time critical CPCIs for the Executive are as follows:

CPCI	NAME	FUNCTION
S115E03	P1SSEC	Phase 1 Control
S115E04	STMBLD	Builds Technology Models
S115E05	COMPLR	Produces Final SSM
S115E07	SCENBD	Builds a Model Scenario
S115E14	DBM	Data Base Manager
S115E10	BATCH	Handles Batch Data

Using program stubs and special test devices, these CPCIs could produce a testable model. The time critical aspects of the schedule are predicated on the schedule of the above programs.

6.2.2 Critical items - technology. Critical technology items for development for the Level One Release include:

(1) Rotor blade finite element model for blade eigenvalue/eigenvector solution (and possibly blade forced response)

(2) Numerical Methods Library - particularly efficient numerical integration schemes

The blade structural dynamic analysis model is critical to the entire rotor analysis. Efficiency of numerical integration schemes has a significant impact on computer use cost and the trade that results between cost and accuracy. Critical technology items for development for the Level Two Release include:

(3) Component mode coupler

(4) Constant coefficient and periodic coefficient aeroelastic stability analyses

The component mode coupler (component mode synthesis) approach is considered an important part of the proposed technology since it permits generality and flexibility that provides for varying levels of complexity.

Constant coefficient and periodic coefficient analysis capability for aeroelastic stability analysis need to be developed to the point where the System's capability for determining constant and periodic coefficient matrix coefficients from a helicopter model (SSM) can be demonstrated.

7. GLOSSARY[9]

Acceptance Testing - Testing to verify that a contractor or subcontractor has met the requirements of the functional specifications applicable to his contract for development of a portion of the CHAS.

Aircraft Life Cycle Phase - A period during the total life cycle of an aircraft or component during which analysis is performed for the purpose of satisfying various levels of requirements. The three phases of concern for this effort are:

1. Research. That phase of the aircraft life cycle during which information is accrued solely for possible later benefit. Characteristics of the research aircraft life cycle phase include: accuracy of computer results is much more important than economy, relatively few flight conditions are analyzed, the emphasis is on engineering understanding of physical phenomena, input data are generally available in great detail for the aircraft system (or subsystem) being studied, empiricism in the analysis methods is kept to a minimum.

2. Preliminary Design. That phase of the aircraft cycle during which an in-depth study of selected alternative configurations to refine their shape, structure, and systems and to improve prediction of performance, weights, and costs. Characteristics of the preliminary design aircraft life cycle phase include: accuracy in trends with economy is more important than absolute accuracy of computer results, a very large number of flight conditions are analyzed for a number of configurations, the emphasis is on selecting an optimum baseline configuration for detailed design, input data are not generally available in great detail, methods need to be general enough to be applicable to the wide variety of configurations that are investigated.

3. Detailed Design. That phase of the aircraft life cycle during which the design of one vehicle configuration is verified and refined and all the technical details are defined preparatory to fabrication. Characteristics of the detailed design aircraft life cycle phase include: accuracy is more important than economy, a large number of flight conditions are analyzed, many of which are near the boundary of the flight envelope or are otherwise critical to the

---

[9]NOMENCLATURE FOR THE SECOND GENERATION COMPREHENSIVE HELICOPTER ANALYSIS SYSTEM, Applied Technology Laboratory, U.S. Army Research and Technology Laboratories (AVRADCOM), Fort Eustis, Virginia.

success of the design, the emphasis is on producing a prototype aircraft which will require a minimum number of changes to meet all performance and structural requirements, detailed input data are developed for the entire aircraft, and empiricism in the analysis methods is often justified by economy and accuracy considerations.

Aircraft Technical Characteristics - Performance, stability and control, loads, acoustics, and aeroelastic stability characteristics.

CHARLES - Comprehensive Helicopter Analysis Language for Engineer Studies; the user language for CHAS.

Computer Program Configuration Item (CPCI) - A software system capable of being handled as a separate entity and unique configuration.

Detailed Functional Capability - Corresponds to a single combination of aircraft technical characteristics, life cycle phase, aircraft configuration, maneuver and operating conditions, and failure damage effects defined in the Type A system specification.

First Level System Release - First release of the System for use by Government and industry, after approximately 2 years of the 4-year System development program.

General Functional Capability (GFC) - The complete input, output, and processing capabilities of the Second Generation Comprehensive Helicopter Analysis System.

Input Group - A uniquely identified set of data representing potential input for a specific function (Software Module) within a Technology Module. An input group may be one or more scalar items or an array.

Model Builder - A person who constructs Specific Technology Modules from Technology Modules and Scenarios or Specific Scenarios to develop Specific Simulation Models.

Model User - A person who uses a Specific Simulation Model to analyze a particular helicopter engineering problem.

Narrative (System-defined) - A free text description of a Technology Module or a Software Module component of a Technology Module that is defined by the Technology Module developer.

Narrative (User-defined) - A free text description of a Specific Technology Module or a software module component of a Technology Module, Specific Simulation Model, or a Specific Scenario, which is defined by the Model User as part of the model building process for a particular Specific Technology Module, Specific Scenario, or Specific Simulation Model.

Output Group - A uniquely identified set of data associated with one Technology Module. An output group may consist of one or more scalar items, an array, or an array plus one or more scalar items. It represents potential output from a Technology Module.

Particular Functional Capability (PFC) - Each PFC corresponds to a Detailed Functional Capability or a set of Detailed Functional Capabilities composed of similar analysis tasks. Each PDC will have a predetermined logic path through the System enabling the model user to perform the specific analysis without having to rely on the GFC.

Program - A subdivision of a software segment or in some cases a subsystem; the lowest level for which formal documentation is prepared.

Quality Assurance - The process of establishing requirements and performing tests to ensure that high level objectives of accuracy, efficiency, and acceptance of the system are met.

Scenario - The defined flow among the Specific Technology Modules (including Specific Scenarios) for a Specific Simulation Model.

Second Level System Release - The second major release of the System for use by Government and industry; at the end of a major development effort spanning approximately 4 years (approximately 2 years after the First Level System Release).

Software Module - Sections of computer program code that represent the smallest subdivision of a complete technological representation of a helicopter component, phenomenon, or solution technique.

Software Segment - A subdivision of a subsystem. Software segments are defined for administration and documentation requirements but also represent a collection of related functions.

Specific Scenario (SS) - A named and stored scenario that may be used as a building block in another scenario.

Specific Simulation Model (SSM) - A simulation model (Specific Technology Modules), built by the model building system specifically for an engineering study.

Specific Technology Module - Represents a set of software modules chosen by the Model Builder applicable to the level of complexity of the subject engineering problem.

Subsystem - A major subdivision of the Second Generation Comprehensive Helicopter Analysis System. The subsystems are the Executive and the Technology Modules.

System - The entire Second Generation Comprehensive Helicopter Analysis System (CHAS).

Technology Integration Contractor - A helicopter manufacturer who acts as a subcontractor to the Primary Development Contractor (PDC); develops requirements and specifications for Technology Modules for the Technology Module developers; monitors development of Technology Modules; acts as an interface between Technology Module developers and the Prime Development Contractor.

Technology Module (TM) - Contains the most complete representation of a particular technology including mutually exclusive processing paths.

Technology Module Developer (TMD) - A subcontractor who develops a particular Technology Module for incorporation into the Second Generation Comprehensive Helicopter Analysis System.

Testing - The systematic checking of the Executive and Technology Modules to see that functional requirements of specifications are met.

Type A System Specification - The System specification for the Second Generation Comprehensive Helicopter Analysis System. The context and format of this document is established by MIL-STD-490.

Type B Development Specification - The system performance specifications used for the procurement of the Technology Modules.

Unit Testing - Testing of a small portion of a computer system to ensure that it performs as designed.

Validation Testing - Testing to determine the accuracy which a Technology Module or Specific Simulation Model produces relative to physical test data for specific Particular Functional Capabilities or Detailed Functional Capabilities.

8. REFERENCES

1. Military Standard, MIL-STD-490, SPECIFICATION PRACTICES, Department of Defense, Washington, D.C., 30 October 1968.

2. DoD Manual 4120.17M, AUTOMATED DATA SYSTEM DOCUMENTATION STANDARDS MANUAL, Department of Defense, Washington, D.C., December 1972.

3. BASELINE TYPE A SYSTEM SPECIFICATION, SECOND GENERATION COMPREHENSIVE HELICOPTER ANALYSIS SYSTEM, Science Applications, Inc. and Boeing Vertol Company, 20 January 1978.

4. BASELINE DEVELOPMENT PLAN, SECOND GENERATION COMPREHENSIVE HELICOPTER ANALYSIS SYSTEM, Science Applications, Inc. and Boeing Vertol Company, 7 March 1978.

5. INTERIM TECHNICAL REPORT - TASK III ANALYSIS, SECOND GENERATION COMPREHENSIVE HELICOPTER ANALYSIS SYSTEM, Science Applications, Inc. and Boeing Vertol Company, 7 March 1978.

6. TYPE B5 DEVELOPMENT SPECIFICATIONS, SECOND GENERATION COMPREHENSIVE HELICOPTER ANALYSIS SYSTEM, Science Applications, Inc. and Boeing Vertol Company, 22 May 1978.

7. Hurty, W.C., DYNAMIC ANALYSIS OF STRUCTURAL SYSTEMS USING COMPONENT MODES, AIAA Journal, Vol. 3, No. 4, April 1965, pp. 678-684.

8. PROGRAMMING STANDARDS FOR THE SECOND GENERATION COMPREHENSIVE HELICOPTER ANALYSIS SYSTEM, in preparation by Applied Technology Laboratory, U.S. Army Research and Technology Laboratories (AVRADCOM), Fort Eustis, Virginia.

9. NOMENCLATURE FOR THE SECOND GENERATION COMPREHENSIVE HELICOPTER ANALYSIS SYSTEM, Applied Technology Laboratory, U.S. Army Research and Technology Laboratories (AVRADCOM), Fort Eustis, Virginia.

APPENDIX

EXAMPLES OF SPECIFIC SIMULATION MODELS

This appendix provides examples of Specific Simulation Models (SSM), built to solve typical problems. These examples illustrate:

- Initializing
- Building a Specific Scenario (SS) from other Specific Scenarios
- Using Specific Technology Modules (STM) with different levels of complexity
- Obtaining a steady state response
- Obtaining a transient response
- Calculating aeroelastic stability
- Downwash coupling with other components
- Rotor/fuselage coupling
- Multirotor problem with mismatched blades

Each example will show the CHARLES language code, the SSM flow diagram, and a brief narrative for each STM and SS used to define the SSM.

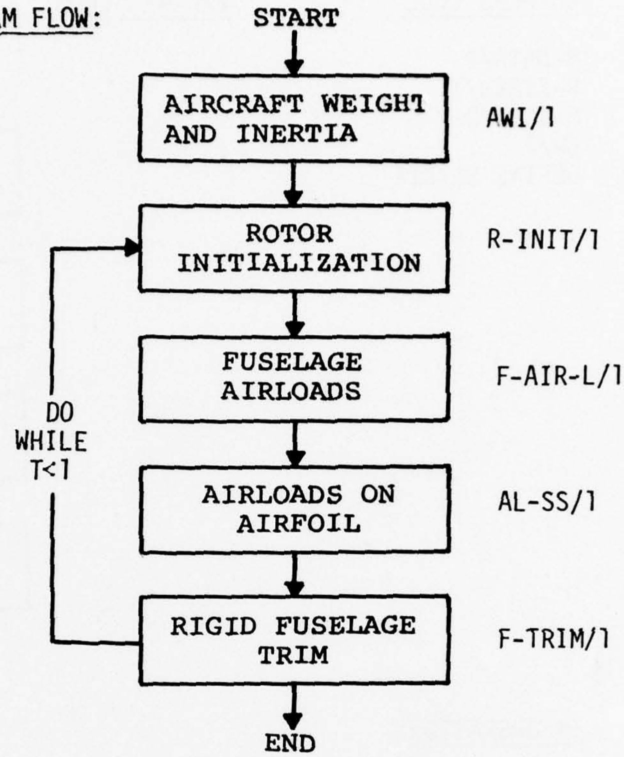
It should be noted that, in gross terms, the flow diagram for the steady state analysis (using undetermined coefficients) and the transient analysis (initial value problem) are very similar. The difference is that for the steady state response, one DO WHILE loop corresponds to one complete rotor revolution, while for the transient response one DO WHILE loop corresponds to one azimuth increment (where azimuth increment is proportional to $\Delta t/h$ and h is the order of the numerical method).

STEP 1 - FUSELAGE INITIAL TRIM

<u>CHARLES CODE</u>

```
    AWI/1
    DO 130 WHILE T.LT. 1
    R-INIT/1
    F-AIR-L/1
    AL-SS/1
130 F-TRIM/1
    DEFINE SS/TRIM 1
```

<u>PROGRAM FLOW:</u>



<u>TM OPERATIONS</u>

<u>AWI/1</u> - STM built from TM25. Calculates aircraft rigid body weight and inertia.

<u>R-INIT/1</u> - STM built from TM09. Calculates steady hub loads with linear aerodynamics and uniform downwash. Uses a rigid articulated blade with a flap spring.

<u>F-AIR-L/1</u> - STM built from TM12. Calculates airloads on the fuselage including stores. The effect of rotor uniform downwash is included.

<u>AL-SS/1</u> - STM built from TM07. Calculates steady airloads on the empennage.

<u>F-TRIM/1</u> - STM built from TM11. Determines fuselage equilibrium orientation. Controls trim match by setting T. Uses an elementary tail rotor.

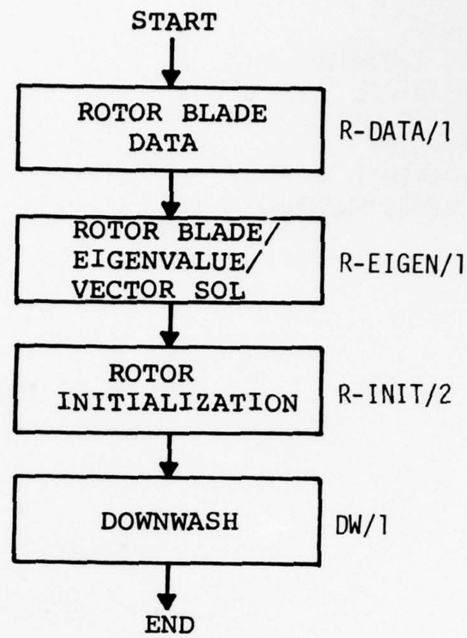
<u>SS/TRIM 1</u> - A Specific Scenario defined by the above CHARLES language commands. Calculates fuselage trim attitudes and determines the main and tail rotor hub loads required to maintain trim.

STEP 2 - INITIALIZE ROTOR

CHARLES CODE

R-DATA/1
R-EIGEN/1
R-INIT/2
DW/1
DEFINE SS/RI1

PROGRAM FLOW:



TM OPERATIONS

R-DATA/1 - STM built from TM01. Converts distributed parameter data into lumped parameter form.

R-EIGEN/1 - STM built from TM02. Calculates blade eigenvalues and eigenvectors. Calls eigen routine from math package. Finds six coupled flag-lay-pitch modes.

R-INIT/2 - STM built from TM09. Solves for the steady state blade response using linear aerodynamics and three coupled modes. Matches required steady hub loads. Calls matrix inversion from math pack.

DW/1 - STM built from TM06. Calculates simplified "skewed" downwash (i.e., linear variation).

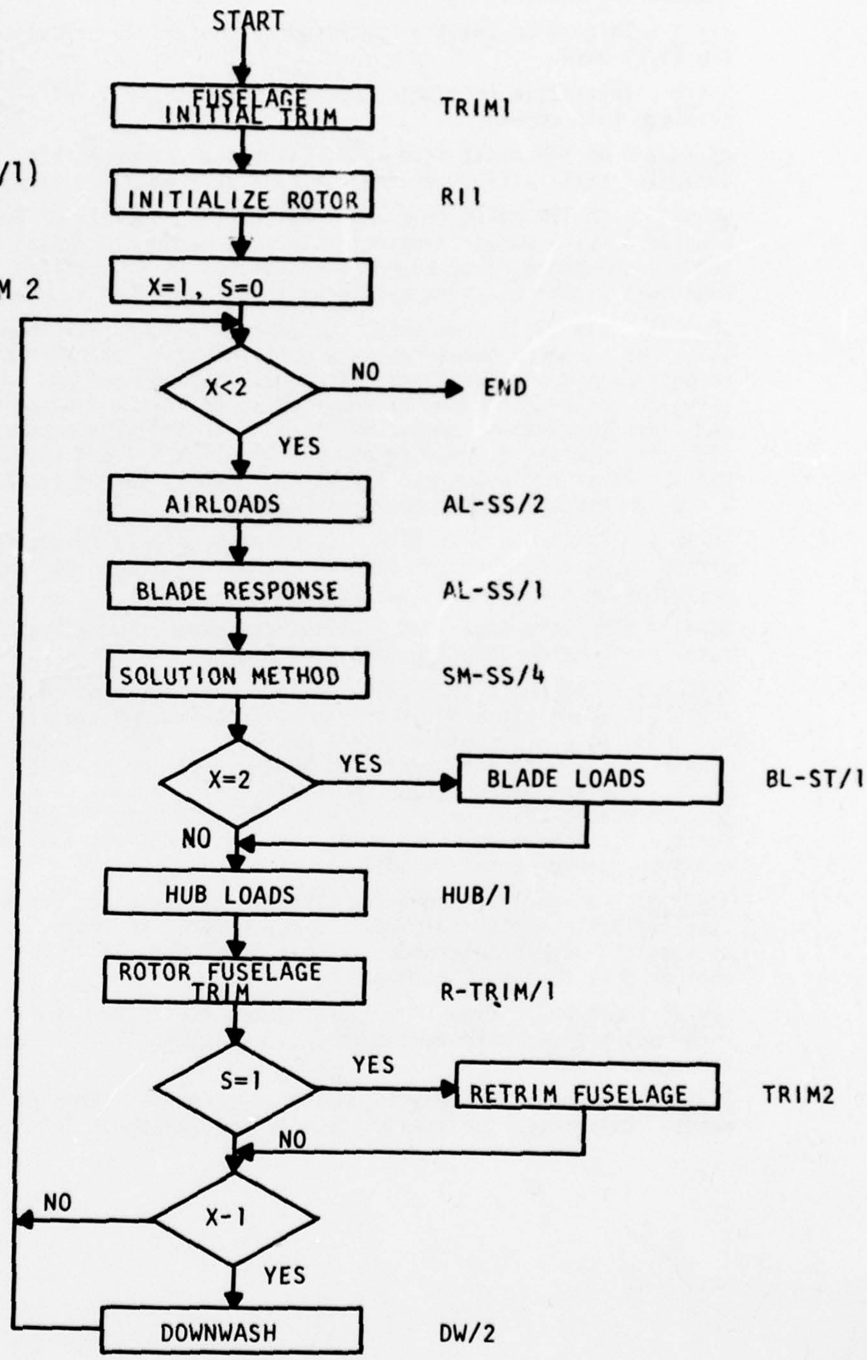
SS/RI1 - A Specific Scenario defined by the above CHARLES language commands. Initializes the rotor blade deflections, airloads and downwash for a rotor with specific hub loads.

STEP 3 - STEADY STATE TRIM ANALYSIS WITH FLEXIBLE BLADES AND RIGID FUSELAGE

<u>CHARLES CODE</u>

```
    SS/TRIM 1
    SS/RI1
    X = 1
    S = 0
    DO 110 WHILE X LT.2
    AL-SS/2
    BR-SS/1
    IF X = 2, THEN (BL-ST/1)
    ELSE CONTINUE
    HUB/1
    R-TRIM/1
    IF S = 1, THEN SS/TRIM 2
    ELSE CONTINUE
    IF X = 1, THEN (DW/2)
    ELSE CONTINUE
110 CONTINUE
    DEFINE SS/RSS1
```

<u>PROGRAM FLOW</u>



TM OPERATION

SS/TRIM 1 - A Specific Scenario calculates the fuselage initial trim.

SS/RI1 - A Specific Scenario calculates blade modes, generalized mass, initial rotor deflections, airloads and downwash (using linear aerodynamics). Matches the required hub loads.

X = 1 - Initialize solution convergence control to calculate downwash on the first pass.

S = 0 - Initialize trim match control (if set S = 1, will do an initial fuselage trim check)

AL-SS/2 - An STM built from TM07. Calculates compressible, unsteady airloads including stall effects on the rotor blade around the azimuth.

BR-SS/1 - An STM built from TM03. Calculates generalized force for six fully coupled (F-L-P) modes in harmonic form, assuming identical blades, and applies the appropriate boundary conditions (i.e., defines the required equations within matching cyclic pitch or hub moments, etc.)

SM-SS/1 - STM built from TM15. Calls matrix inversion from math pack to solve for harmonic modal response coefficients. Calculates blade deflections, velocities and accelerations. Includes loop counter and performs tests for solution convergence (can be required to perform a minimum number of loops and limit the maximum number of loops). If solution converged, set X = 1 (to calculate the nonuniform downwash). If X = 1 and solution converged, set S = 2; if not converged set X = 0. (NOTE: Cannot set X = 2 unless S = 2, or maximum loop number reached).

BL-ST/1 - STM built from TM18. Calculates internal blade loads and stress using the force integration method. ( ) used to identify this STM as secondary for storage considerations.

HUB/1 - STM built from TM04. Calculates steady and vibratory hub loads and rotor performance assuming identical blades.

R-TRIM/1 - STM built from TM05. Adjust collective pitch and rotor shaft angle (fuselage pitch attitude) to match trimmed hingeless rotor steady hub loads (i.e., propulsive force and lift). For a hingeless rotor, hub moment is automatically satisfied by the equation in BR-SS/1. If trim rotor loads are matched, but the shaft angle, torque or side force exceed the acceptable tolerance, set S = 1. If trim rotor loads are matched and shaft angle change is acceptable, set S = 2 (and the loop may be exited if the solution method is satisfied).

SS/TRIM2 - A specific Scenario. Retrim the fuselage for the required fuselage pitch angle and calculate new rotor trim loads. If resulting trim changes are within tolerance, set S = 0, if not, leave S = 1 and perform another pass through the loop.

(DW/2) - STM built from TM06. Calculates nonuniform downwash on the main rotor using rigid wake vortex theory. ( ) identify as a secondary STM for storage.

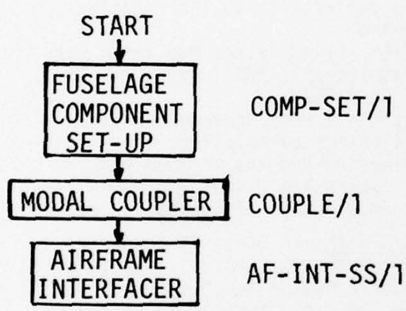
SS/RSS1 - A Specific Scenario defined by the above CHARLES language commands. Calculates the steady state trimmed response for flexible blades.

STEADY STATE ANALYSIS, OBTAIN FUSELAGE MOBILITIES FOR A SINGLE ROTOR

CHARLES CODE

COMP-SET/1
COUPLE/1
AF-INT-SS/1
DEFINE SS/F-START

PROGRAM FLOW



TM OPERATIONS

COMP-SET/1 - STM built from TM 14 collects all component mode data and organizes it for the modal coupler, including location, boundary conditions, etc.

COUPLE/1 - STM built from TM 13B sets up the equations for the combined (coupled) fuselage system. Calculates the new eigenvalue/vectors using routines from the math pack. Selects the desired sub set of modes.

AF-INT-SS/1 - STM built from TM 13A calculates fixed system fuselage mobilities at the hub for the necessary frequencies.
NOTE: STM can be built to calculate mobilities and cross mobilities for up to four hubs

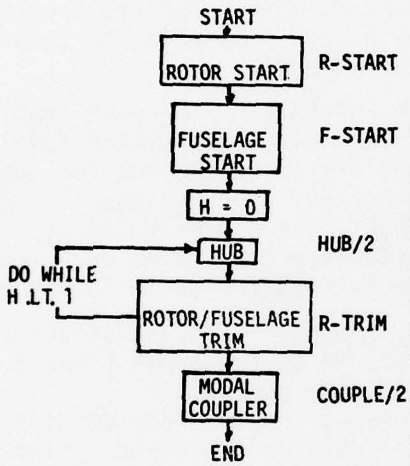
SS/F-START - A Specific Scenario defined by the above CHARLES language commands.
Sets up the fuselage modal equation and defines the fixed system hub mobility.

STEADY STATE ANALYSIS - CALCULATE ROTOR LOADS, TRIM, PERFORMANCE AND FUSELAGE VIBRATION FOR A SINGLE FLEXIBLE ROTOR AND FUSELAGE

CHARLES CODE

```
    SS/R-START
    SS/F-START
    H = 0
    DO 120 WHILE H.LT. 1
    HUB/2
    SS/R-TRIM
120 CONTINUE
    COUPLE/2
```

PROGRAM FLOW



TM OPERATIONS

SS/R - START - A Specific Scenario. calculates all blade modal data, performs initial trim and initializes rotor loads, deflections, and downwash.

SS/F-START - A Specific Scenario. Sets up the fuselage modal equations and defines the fixed system hub mobility.

H = 0 - Initializes hub coupling loop.

HUB/2 - STM built from TM04. Transfer appropriate mobilities into the rotating system (example: inplane mobility is not transferred since forcing is non linear). Calculate deflections in the fixed system for those mobilities that were not transferred, and transfer the deflections to the rotating system.
NOTE: Fixed system hub loads are calculated in HUB/1 which is part of SS/R-TRIM.
Set H = 1 when consecutive hub deflection calculations are within tolerance and the minimum number of cycles are exceeded or when maximum number of cycles is reached.

SS/R-TRIM - A Specific Scenario. Calculates the steady state trimmed rotor response for flexible blades and the rigid fuselage response.
NOTE: The rotor blade equations of motion include rigid body hub degrees of freedom. Degrees of freedom that use the rotating system mobilities automatically couple with the fuselage. Nonlinear terms or a hub with mismatched blades require iteration between HUB/2 and SS/R-TRIM, controlled by H to achieve convergence (compatibility or coupling).

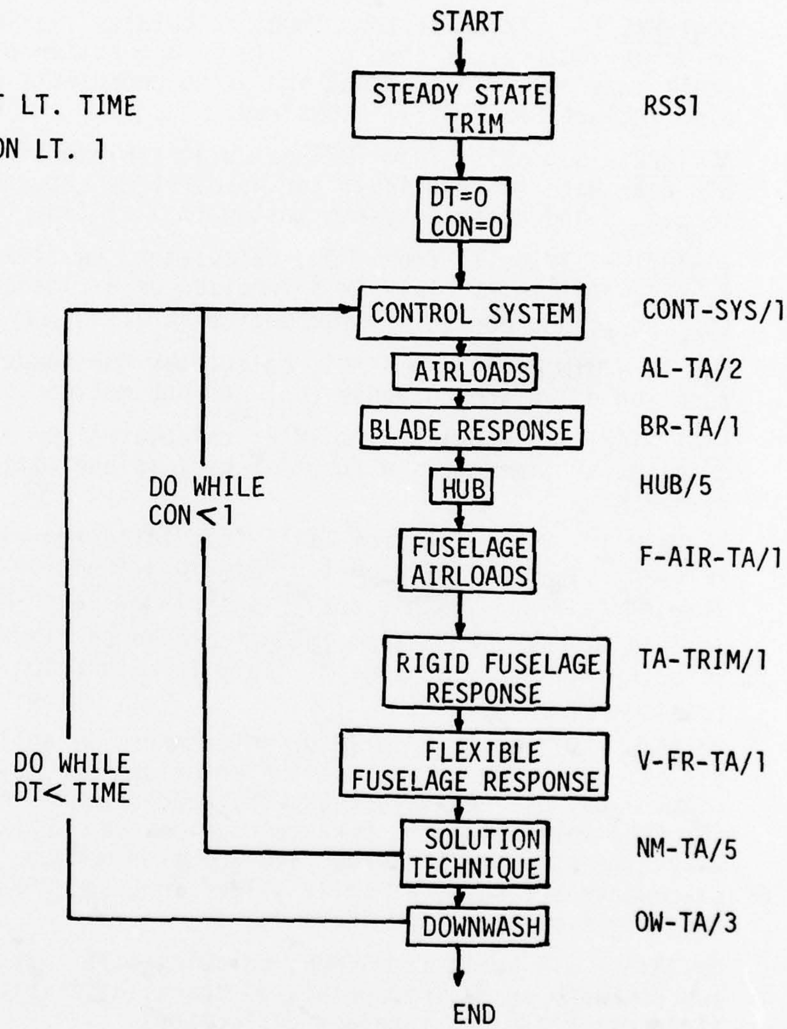
COUPLE/2 - STM built from TM13B. Use the final (converged) fixed system hub loads to calculate fuselage vibration at indicated coordinates using the equations for the combined fuselage system (matrix equations), i.e., Couple back into the equations built up in COUPLE/2.

TRANSIENT ANALYSIS WITH FLEXIBLE BLADES AND FUSELAGE

CHARLES CODE

```
    SS/RSS1
    DT=0
    CON=0
    DO 150 WHILE DT LT. TIME
    DO 100 WHILE CON LT. 1
    CONT-SYS/1
    AL-TA/2
    BR-TA/1
    HUB/5
    F-AIR-TA/1
    TA-TRIM/1
    V-FR-TA/1
100 NM-TA/5
    DW-TA/3
    CON=0
150 CONTINUE
```

PROGRAM FLOW



TM OPERATIONS

SS/RSS1 - A Specific Scenario calculates the steady state trimmed response for a rotor with flexible blades. Defines the initial conditions for the problem.

DT=0, CON=0 - Initialize control loops.

CONT-SYS/1 - STM built from TM08; calculates flexible control system velocity, $\dot{q}c$=( ), at time t. This is a function of pilot input (call interpolation routine) and blade coordinate response (q_B). Can also include SAS (TM12) if desired.

AL-TA/2 - STM built from TM07 calculates blade airloads as a function of: near wake (in routine), for wake, blade response (q_B), hub motion (q_{FR},q_{FF}) and control system motion (q_c) at time t.

BR-TA/1 - STM built from TM03, calculates the flexible blade velocity, $\dot{q}_B$=( ), at time t. This is a function of airloads, hub motion (q_{FR}, q_{FF}) and control system motion (q_c).

HUB 15 - STM built from TM04; calculates hub loads at time t, as a function of: blade response (q_B) and hub motion (q_{FR}, q_{FF}).

F-AIR-TA/1 - STM built from TM12; calculates the airloads on the fuselage at time t, as a function of fuselage motion, and far wake downwash.

TA-TRIM/1 - STM built from TM11; calculates the rigid body fuselage velocity, $\dot{q}_{FR}$=( ), at time t. This is a function of hub loads and fuselage airloads. This analysis includes large angular deflections.

V-FR-TA/1 - STM built from TM13A; calculates flexible fuselage velocity velocity, $\dot{q}_{FF}$=( ), at time t. This is a function of hub loads and fuselage airloads.

NM-TA/5 - STM built from TM15; solves for all deflections (q) at time t + Δt. Uses CON to control the solution loop, by looping until the routine has all the velocities ($\dot{q}$) needed consistent with the solution method order. Certain routines can perform error checks, vary time step, method order and solution method. When the displacements are satisfactorily solved for, set CON=1 to continue to the next time step, and increment T.

DW-TA/3 - STM built from TM06; calculates far wake vortex strength and downwash on desired points in space, as a function of blade airloads, blade and fuselage deflections.

STEADY STATE ANALYSIS - CALCULATE ROTOR LOADS, TRIM, PERFORMANCE AND FUSELAGE VIBRATION FOR AN AIRCRAFT CONFIGURATION WITH 2 ROTORS AND MISMATCHED BLADES ON EITHER ROTOR

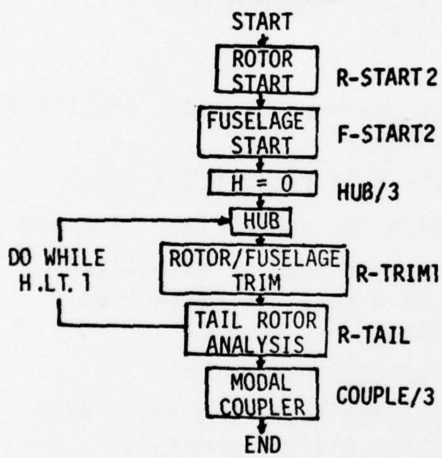
- Use identical analysis method for the two rotors.
  - The CHARLES SSM will look almost the same
  - Appropriate STM's will be modified to include two additional variable dimensions (i.e., blade and rotor)
  - When a STM is entered to calculate values for one blade of one rotor, all blades in all rotors will be analyzed.
  - STM modification will include:
    - Downwash will be calculated on tail surfaces, and each blade of each rotor
    - HUB/2 includes cross mobility matrix
    - HUB/1 includes mismatched blades
    - All rotor associated STM will include subscripts for blade and rotor.
- Use the above analysis for the main rotor and a simplified analysis for the tail rotor.
  - Define an SS to analyze the tail rotor (SS/R-TAIL)
  - Identify the rotor data with the appropriate STMs and SSs
  - Main rotor downwash will be modified to calculate discrete vortex wake on: main rotor (self induced), tail surface and tail rotor
  - Tail rotor downwash will be uniform
  - Tail rotor will be a two-bladed teetering rotor with linear aerodynamics and three elastic modes.

STEADY STATE ANALYSIS - CALCULATE ROTOR LOADS, TRIM, PERFORMANCE AND FUSELAGE VIBRATION FOR AN AIRCRAFT CONFIGURATION WITH 2 ROTORS (WITH DIFFERENT ANALYTICAL SOLUTION TECHNIQUES) AND MISMATCHED BLADES IN EITHER ROTOR

CHARLES CODE

```
    SS/R-START2
    SS/F-START2
    HUB = 0
    DO 120 WHILE H.LT.1
    HUB/3
    SS/R-TRIM 1 (*) R-1
    SS/R-TAIL (*) R-2
120 CONTINUE
    COUPLE/3
```

PROGRAM FLOW



TM OPERATIONS

SS/R-START 2 - A Specific Scenario. Calculates all requested modal data for Rotor 1 (R-1) the main rotor and Rotor 2 (R-2) the tail rotor. Performs initial trim using uniform downwash and initializes rotor loads, deflections, and main rotor skewed downward.

SS/F-START 2 - A Specific Scenario. Sets up the fuselage modal equations and defines the fixed system hub mobility and cross mobility for both R-1 and R-2.

H = 0 - Initializes hub coupling loop.

HUB/3 - STM built from TM04. Calculates deflection in the fixed system for mismatched blades for both rotors, including cross mobility effects. Controls loop by setting H = 1 when complete.

(*) R-1 - This indicates that the Specific Scenario SS/R-TRIM1 uses data from the main rotor only.

SS/R-TRIM 1 - A Specific Scenario. Calculates the steady state trimmed rotor response for mismatched, flexible main rotor blades and the rigid fuselage.

SS/R-TAIL - A Specific Scenario. Calculates the steady state trimmed rotor response for mismatched, flexible, two-bladed teetering tail rotor, using linear aerodynamics and uniform self-induced downwash, and main rotor interference downwash.

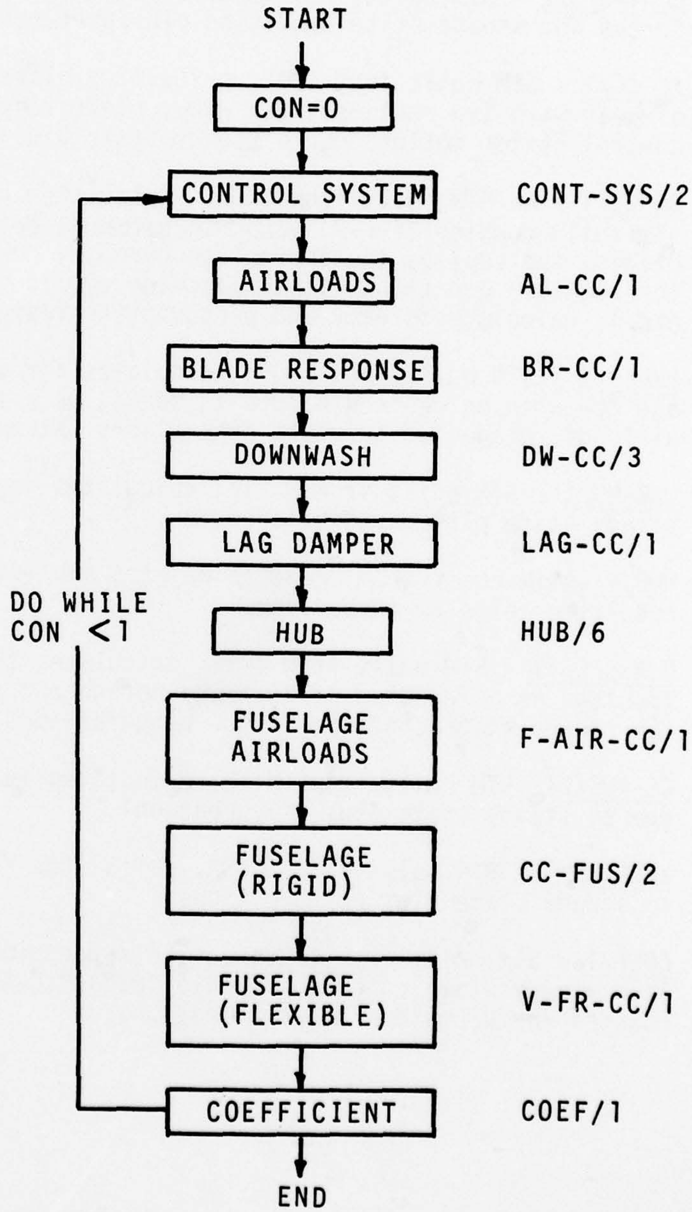
COUPLE/3 - STM built from TM13B. Uses the final (converged) fixed system main and tail rotor hub loads to calculate fuselage vibration at indicated coordinates.

COEFFICIENT DETERMINATION FOR AEROELASTIC STABILITY ANALYSIS SPECIFIC SCENARIO SS/CO-AN1

CHARLES CODE FOR SS/CO-AN1

```
    CON=0
    DO 100 WHILE CON .LE. 1
    CONT-SYS/2
    AL-CC/1
    BR-CC/1
    DW-CC/3
    LAG-CC/1
    HUB/6
    F-AIR-CC/1
    CC-FUS/2
    V-FR-CC/1
100 COEF/1
    DEFINE SS/CO-AN1
```

PROGRAM FLOW - SS/CO-AN1



COEFFICIENT DETERMINATION FOR AEROELASTIC STABILITY ANALYSIS SPECIFIC SCENARIO SS/CO-AN1

TM OPERATIONS

SS/RSS1 - A Specific Scenario; calculates trimmed response for a rotor with flexible blades, defines the initial conditions for the problem.

CON=0 - Initializes control loop.

CONT-SYS/2 - STM built from TM08; calculates flexible control system forces for steady state condition plus perturbation.

AL-CC/1 - STM built from TM07; calculates blade airfoils as a function of near wake (in routine), far wake, blade response, hub motion, and control system motion due to steady state plus perturbations.

BR-CC/1 - An STM built from TM03. Calculates generalized force for six fully coupled (F-L-P) modes in harmonic form, assuming identical blades, and applies the appropiate boundary conditions (i.e., defines the required equations within matching cyclic pitch or hub moments, etc.); calculates steady and perturbation response.

DW-CC/3 - STM built from TM06; calculates far wake vortex strength and downwash on desired points in space, as a function of blade airloads, blade and fuselage deflections (steady state plus perturbations).

LAG-CC/1 - STM built from TM 16; calculates lag damper forces due to steady state plus perturbations.

HUB/6 - STM built from TM04; calculates hub loads due to steady state conditions plus perturbations.

F-AIR-CC/1 - STM built from TM12; calculates the airloads on the fuselage as a function of fuselage motion and far wake downwash for steady state conditions plus perturbations.

CC-FUS/2 - STM built from TM11B; calculates fuselage rigid body forces due to steady state plus perturbations.

V-FR-CC/1 - STM built from TM13A; calculates fuselage mode forces due to steady state plus perturbations.

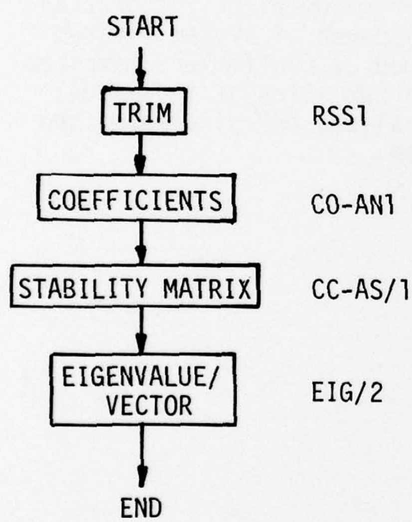
COEF/1 - STM built from TM28; controls perturbation of state variables from steady state conditions; calculates M,C,K matrix coefficients from changes in forces due to perturbations.

CONSTANT COEFFICIENT AEROELASTIC
STABILITY ANALYSIS

CHARLES CODE

SS/RSS1
SS/CO-AN1
CC-AS/1
EIG/2

PROGRAM FLOW:



TM OPERATIONS

SS/RSS1 - A Specific Scenario; calculates trimmed response for a rotor with flexible blades; defines the initial conditions for the problem.

SS/CO-AN1 - A Specific Scenario; calculates coefficient matrices of equations of motion for a specific rotor azimuth.

CC-AS/1 - STM built from TM28; performs quasi-normal transformation of rotor blade coordinates, develops stability matrix from transformed coefficient matrices.

EIG/2 - STM built from TM15; computes complex eigenvalues and eigenvectors of stability matrix.

SPECIFIC SCENARIO FOR FLOQUET ANALYSIS - SS/FLOQ1

CHARLES CODE - SS/FLOQ1

```
      CON=0
      DO 300 WHILE CON .LT.1
      PC-AS/1
300   SM/3
      DEFINE SS/FLOQ1
```

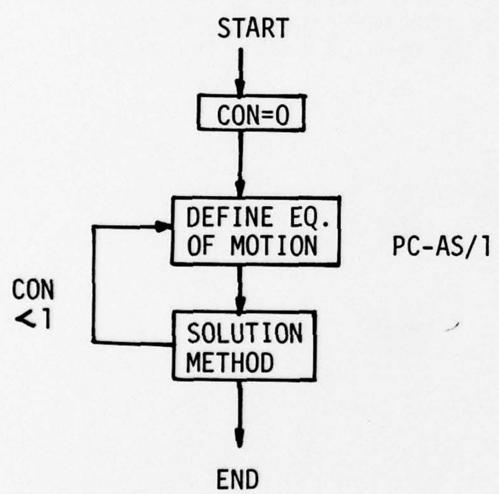
TM OPERATIONS

CON - Controller for loop.

PC-AS/1 - STM built from TM28; periodic coefficient matrices are used to define equations of motion for numerical integration; each generalized coordinate is perturbed independently; integration over one rotor period (by SM/3) is used to define relationship of initial perturbation values and final values to give FLOQUET transition matrix.

SM/3 - STM built from TM15; performs numerical integration of response of system (system defined by coefficient matrices) to perturbation of individual generalized coordinates of the system.

PROGRAM FLOW

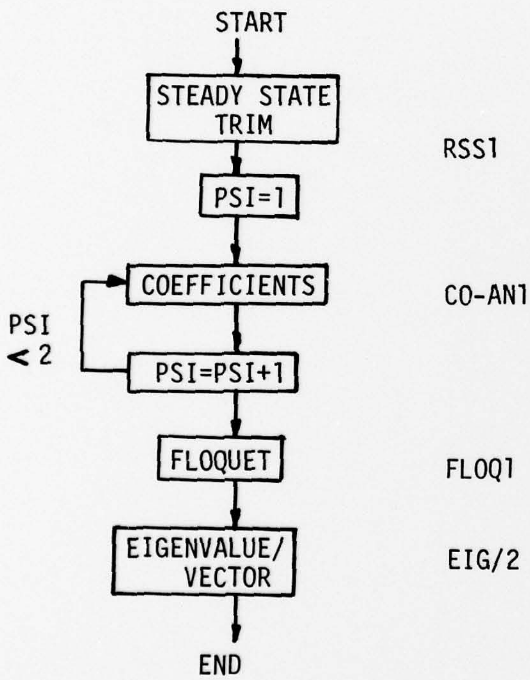


PERIODIC COEFFICIENT AEROELASTIC STABILITY ANALYSIS

CHARLES CODE

```
     SS/RSS1
     PSI=1
     DO 100 WHILE PSI .LE.2
200  SS/CO-AN1
     SS/FLOQ1
     EIG/2
```

PROGRAM FLOW



TM OPERATIONS

SS/RSS1 - A Specific Scenario; calculates trimmed response for a rotor with flexible blades, defines the initial conditions for the problem.

PSI - Counter for number of azimuths where coefficients are to be determined.

SS-CO-AN1 - A Specific Scenario; calculates coefficient matrices for a given azimuth position of the rotor (0 and 90 degrees).

SS/FLOQ1 - A Specific Scenario; calculates FLOQUET transition matrix from periodic coefficient matrices at several azimuths computed by SS/CO-AN1.

EIG/2 - STM built from TM15; computes complex eigenvalues and eigenvectors from FLOQUET transition matrices.